A LOW-COMPLEXITY APPROACH FOR MOTION-COMPENSATED VIDEO FRAME RATE UP-CONVERSION

A Dissertation
Presented to
The Academic Faculty

by

Salih Dikbaş

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2011

A LOW-COMPLEXITY APPROACH FOR MOTION-COMPENSATED VIDEO FRAME RATE UP-CONVERSION

Approved by:

Dr. Yücel Altunbaşak, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. David V. Anderson
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Biing-Hwang Juang
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Patricio Antonio Vela
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. David M. Goldsman
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: August 10, 2011

To my late parents, Zöhre and Muzaffer Dikbaş;

to my lovely wife, Zeynep Dikbaş;

to my adorable kids, Burak Emre and Zehra Betül Dikbaş.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Prof. Yücel Altunbaşak, for giving me the opportunity to work under his supervision and for his guidance, encouragement, and support during my years at Georgia Tech. I have benefited greatly not only from his technical excellence but also from his integrity, wisdom, sincerity, and friendship.

I want to thank Prof. David V. Anderson, Prof. Biing-Hwang Juang and Prof. Patricio Antonio Vela for kindly serving both on my proposal and dissertation committee. I also thank Prof. David M. Goldsman for serving on my dissertation committee. I thank all my dissertation committee members for being very supportive of my research and their constructive criticism that have appreciably contributed to my dissertation.

I would like to thank the members of the Multimedia Computing and Communications Lab, as well as the members of the Center for Signal and Image Processing for their support and friendship. I especially thank Tarik Arici for his collaboration during the Ph.D. years; having numerous discussions with him on esoteric topics was a joy I'll never forget.

I want to thank all my teachers and professors that I had throughout my education journey both in Turkey and USA.

Most importantly, I want to thank my late dad, late mom, and brothers. Their love, sacrifice, and support significantly contributed to be who I am.

I'd especially like to thank my late father-in-law and mother-in-law, whose love and support have been a huge motivation for me to complete this work. I'll be always indebted to them for their help through the hard times.

I do not know how to explain my deepest appreciation in words for my lovely wife,

Zeynep Dikbas. This work would not be possible without her support, endurance, encouragement, and proofreading. I'm sure she is much more happier than I am that my Ph.D. is finally over. Company of my wife and our kids Burak Emre and Zehra Betül made this duration a pleasant and joyful journey.

And last, but by no means least, I thank the Almighty God for the blessings he continues to bestow upon my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Video frame rate up-conversion is an important issue for multimedia systems in achieving better video quality and motion portrayal. Motion-compensated methods offer better quality interpolated frames since the interpolation is performed along the motion trajectory. In addition, computational complexity, regularity, and memory bandwidth are important for a real-time implementation.

Motion-compensated frame rate up-conversion (MC-FRC) is composed of two main parts: motion estimation (ME) and motion-compensated frame interpolation (MCFI). Since ME is an essential part of MC-FRC, a new fast motion estimation (FME) algorithm capable of producing sub-sample motion vectors at low computational-complexity has been developed. Unlike existing FME algorithms, the developed algorithm considers the low complexity sub-sample accuracy in designing the search pattern for FME. The developed FME algorithm is designed in such a way that the block distortion measure (BDM) is modeled as a parametric surface in the vicinity of the integer-sample motion vector; this modeling enables low computational-complexity sub-sample motion estimation without pixel interpolation.

MC-FRC needs more accurate motion trajectories for better video quality; hence, a novel true-motion estimation (TME) algorithm targeting to track the projected object motion has been developed for video processing applications, such as motion-compensated frame interpolation (MCFI), deinterlacing, and denoising. Developed TME algorithm considers not only the computational complexity and regularity but also memory bandwidth. TME is obtained by imposing implicit and explicit smoothness constraints on block matching algorithm (BMA). In addition, it employs a novel adaptive clustering algorithm to keep the low-complexity at reasonable levels yet

enable exploiting more spatiotemporal neighbors. To produce better quality interpolated frames, dense motion field at the interpolation instants are obtained for both forward and backward motion vectors (MVs); then, bidirectional motion compensation using forward and backward MVs is applied by mixing both elegantly.

CHAPTER I

INTRODUCTION

As humans, we want to interact with each other to communicate and share our experiences. This temptation led us throughout the ages to invent new technologies to better explain ourselves. The invention of the electric generators in the early 1800s paved the way for many electrical equipments that can be used for this purpose. In 1848, inspired from electric telegraph, F. C. Bakewell invented a scanning technique to electrically transmit and record autographic messages [16]. This invention laid the foundations for still image, image sequence transmission and storage. As a result, this and similar inventions not only influenced our society but also inspired newer and better technologies for many years to come.

Advancements in the silicon technology empowered a growth in complexity since its invention in the mid 1900s. Over time, this drive helped replace their analog counterparts with digital solutions. This transformation led to many multimedia products and proliferation of video formats. As a result, various multimedia devices established themselves in our everyday lives. Existence of diverse video formats with different spatial and temporal characteristics made video format conversion an essential technology in multimedia systems.

Video format conversion techniques can be grouped into three parts: spatial, temporal, and spatiotemporal. Spatial format conversion is achieved by scaling; spatiotemporal format conversion by deinterlacing; and temporal format conversion by frame rate conversion, or temporal interpolation. Compared with other two parts, temporal interpolation is more complicated both fundamentally and practically. For example, conversion from 24 Hz to 60 Hz or 120 Hz requires creation of images at new

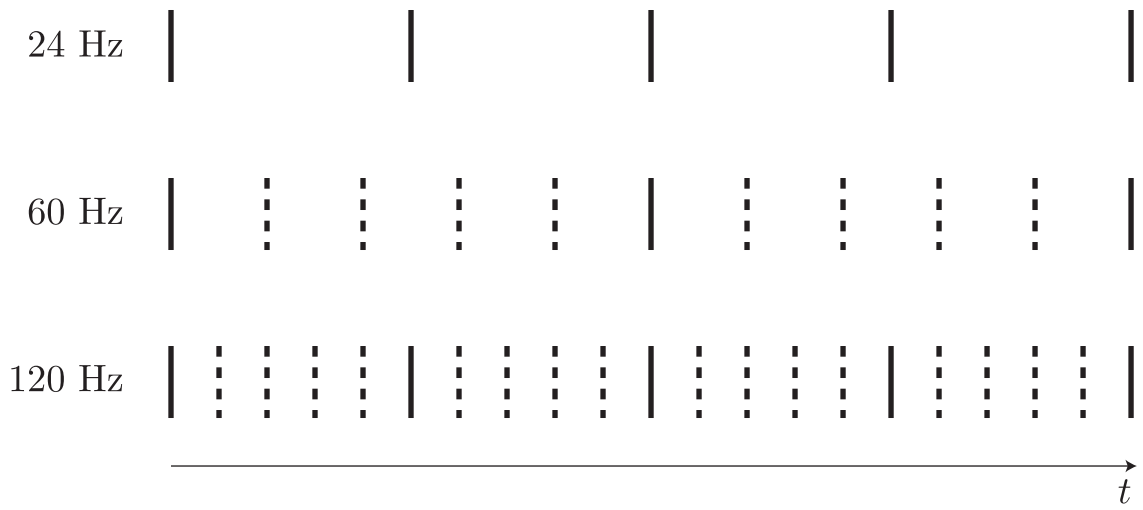


Figure 1: Illustration of interpolation time instants for video format conversion from 24 Hz to 60 Hz or 120 Hz. Solid lines indicate the original frames, dashed lines indicate the interpolated frames.

time instants as illustrated in Figure 1.

Another driving force of temporal interpolation is the inferior dynamic resolution of ubiquitous flat-panel displays (FPDs). Compared with cathode-ray tubes (CRT), FPDs have many advantages: lighter weight, smaller size, lower power consumption, less radiation, and sharper displayed images; hence, FPDs are rapidly replacing CRTs [10]. Due to FPDs' heavy use, people expect them to produce superior image quality. Also, with the added expectation that a newer technology should always improve on its predecessors, the demand for a better picture quality display is at an all-time high.

In recent years, the FPD market has been steadily growing in North America, Asia, and Europe. Most common FPD technologies available in the consumer market today are liquid crystal display (LCD), plasma display panel (PDP), and digital light processing (DLP). They are most commonly used either as TVs or computer monitors. Historically, PDPs were available for screen sizes larger than 40", and LCDs were available for screen sizes smaller than 40". With recent technological advances, however, LCD TVs became available for screen sizes larger than 40" with

competitive market prices. The LCD industry constitutes the majority of the FPD market and is expected to grow at an annual rate of 17% [40]. Despite their appeal, LCDs have a few disadvantages in terms of visual quality that can be addressed with signal processing. They are as follows:

- At nonnative resolutions, LCDs produce blurriness due to scaling
- Artifacts such as noise or compression artifacts in the source video are more visible with larger displays
- The refresh rate at standard frequencies (60 Hz for NTSC, 50 Hz for PAL) causes motion blur and double images
- Large area flicker artifacts become more noticeable in larger displays

For improved visual perception, the input video must be processed before being sent to display. Targeting the first two drawbacks are not covered in this work; however, temporal interpolation is one of the methods used to address the last two drawbacks. Although LCDs are superior to CRTs in many areas, their motion portrayal characteristic is not as good as its antecedent CRTs [35, 48]. Inferior motion portrayal characteristic manifests itself in visual perception either as motion blur at low object speeds or double images at high object speeds.

Motion blur in LCDs is caused by two factors: 1) slow response time of LC material to voltage changes, 2) hold-type character of LCDs. As a solution to the former factor, overdrive techniques are used to compensate for the response time [79, 85]. However, even if the response time could be reduced to zero, the motion portrayal of LCDs would still be inferior to that of CRTs due to the way pixels are displayed; this is because CRTs use impulse-type display and LCDs use hold-type display [35, 48]. As illustrated in Figure 2, in impulse-type displays briefly excited phosphorous results in briefly repeating and immediately decaying images, which are integrated by the

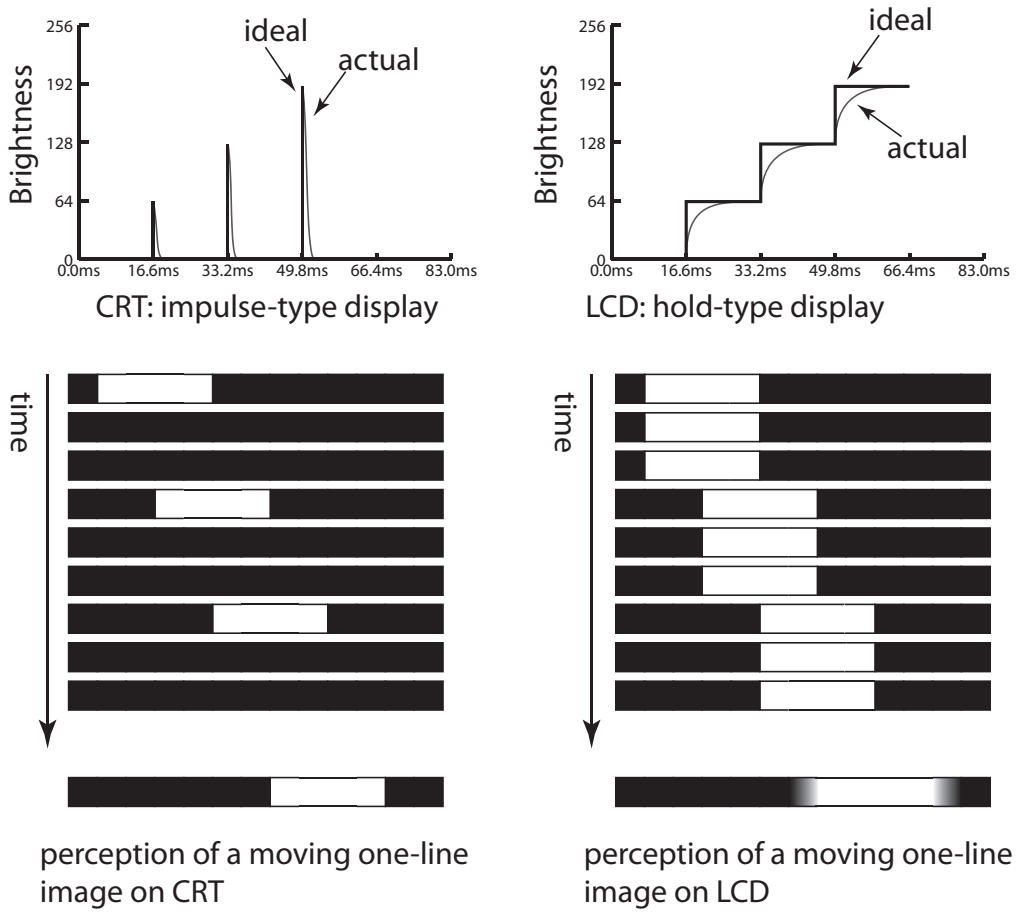


Figure 2: Perception of moving objects on a CRT and an LCD. Adapted from Reference [35].

human visual system (HVS) and perceived as continuous; in hold-type displays, the brightness is continuous and the integration by the HVS results in blur perception at low object speeds and double image perception at high object speeds. Solutions to the latter factor emulate the impulse-type characteristic of CRT. In one solution, black frame is inserted. A better solution is dividing the backlight of panel into several horizontal slices and lighting each sub-panel sequentially during a frame time, which is called *backlight strobing*.

To further improve the quality, LCD manufacturers are increasing the panel frequency, which is usually a multiple of the standard frequency. Although manufacturing LCDs with higher refresh rate improves the perceived quality [35], it does not

completely solve the problem since the source material at these instants is usually missing in the input video. Frame rate of the source video is always kept lower than the refresh rate of the display to reduce transmission and storage costs. Whenever the display rate is different than the source material frame rate, video format conversion has to be utilized; oftentimes, the display rate is the larger one, as a result extra frames have to be generated for display.

Another case requiring the frame interpolation is the use of lower frame rate at the encoder for better utilization of the transmission bandwidth. Some systems encode the source video at a lower frame rate, e.g. by a decimation factor of two or three, to reduce the number of total allocated bits while increasing the number of allocated bits per frame, which potentially leads to a significant gain in picture quality [47, 99]. However, this temporal decimation results in jerky motion at the decoded bitstream. Frame interpolation can be utilized to improve the temporal perceptual quality of the video.

As a result, to attain better motion portrayal, missing frames at the display time instances need to be interpolated from the existing neighboring frames. In the literature, this process is often called frame rate up-conversion (FRC) or frame interpolation. Currently, straightforward and very intuitive methods, such as frame repetition and 2:3 pull-down, are employed for FRC. The successful creation of higher-quality interpolated frames will greatly enhance the motion portrayal and reduce the motion blur. Our goal, therefore, in this work is to develop such an efficient algorithm that is capable of increasing the temporal resolution by creating higher-quality interpolated frames.

1.1 Previous Work

FRC attempts found in the literature can be grouped into two main categories: non-motion-compensated and motion-compensated methods; both categories can further

be divided into two sub-categories: linear and non-linear methods [34]. In the following, we give a brief description of each method.

1.1.1 Non-Motion-Compensated Methods

Methods discussed in this section constitute the methods proposed in the past and may currently still be available in consumer electronics. Although they do not give results as good as motion-compensated methods, we will mention them here for both completeness and a better understanding of the subject. Demo video sequences showing the comparison of picture repetition, picture averaging, and 2:3 pull-down against perfect frame rate up-conversion are shown in Figures 3 and 4.

1.1.1.1 Linear Methods

The methods in this group can be easily expressed in terms of filtering operations performed on one-dimensional signals. For each pixel location, consider a one-dimensional signal composed of pixel values of the previous and future frames at the same location. The signal can extend both to the future and past frames for several frames. Then, the signal is up-sampled by a factor of L, filtered by a temporal filter $h[n]$, and down-sampled by a factor of M as shown in Figure 5. Different sets of L,M, and $h[n]$ result in different linear techniques.

Picture Repetition In this method, the interpolated frame is obtained by repeating the previous frame. In the case of film material (24 Hz) to 60 Hz conversion, alternating frames are repeated once and twice to achieve the desired frame rate. This process is commonly known as 2:3 (or 3:2) pull-down in the literature [42]. Picture repetition results in blurring for objects with small motion vectors and double images for objects with large motion vectors. Also, it introduces motion judder in 2:3 pull-down.

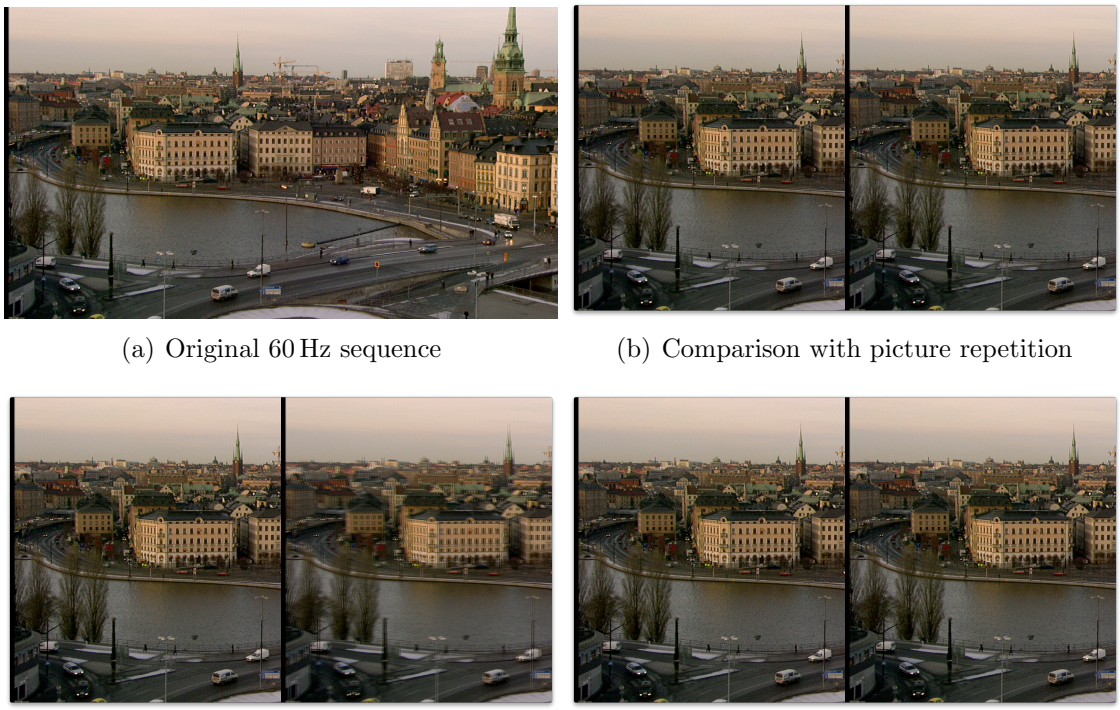
(a) Original 60 Hz sequence

(b) Comparison with picture repetition

(c) Comparison with picture averaging

(d) Comparison with 2:3 pull-down

Figure 3: Demo of perfect FRC against existing non-motion-compensated methods using *movingLetters* video sequence.
(a) (dikbas_salih_201112_phd_movingLetters_org.mov, 151M)
(b) (dikbas_salih_201112_phd_movingLetters_rep.mov, 150M)
(c) (dikbas_salih_201112_phd_movingLetters_ave.mov, 150M)
(d) (dikbas_salih_201112_phd_movingLetters_2-3.mov, 150M)



(a) Original 60 Hz sequence

(b) Comparison with picture repetition

(c) Comparison with picture averaging

(d) Comparison with 2:3 pull-down

Figure 4: Demo of perfect FRC against existing non-motion-compensated methods using *stockholm* video sequence.
(a) (dikbas_salih_201112_phd_stockholm_org.mov, 273M)
(b) (dikbas_salih_201112_phd_stockholm_rep.mov, 271M)
(c) (dikbas_salih_201112_phd_stockholm_ave.mov, 271M)
(d) (dikbas_salih_201112_phd_stockholm_2-3.mov, 271M)

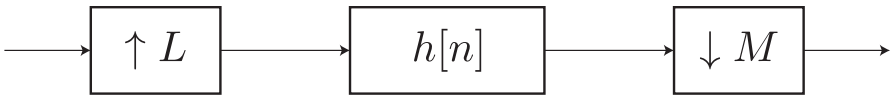


Figure 5: Block diagram for FRC of rate M/L.

Weighted Picture Averaging Weighted picture averaging is analogous to the spatial bi-linear interpolation in the temporal domain. The interpolated frame is obtained from two existing frames at times k and $k+1$ according to

$$I_{k+\alpha,\,avg}[\boldsymbol{x}] = (1-\alpha)I_k[\boldsymbol{x}] + \alpha I_{k+1}[\boldsymbol{x}], \qquad 0 \leq \alpha \leq 1 \tag{1}$$

where $I_k[\boldsymbol{x}]$ is the pixel intensity value at pixel position $\boldsymbol{x} = (x, y)$ of frame k and α is the normalized distance of the interpolated frame from $I_k[\boldsymbol{x}]$.

This method gives very good results in stationary parts. However, edges become blurred in non-stationary regions, where blurring depends on the image velocity and local contrast.

1.1.1.2 Non-Linear Methods

Non-linear methods in this group try to minimize the blurring and double images caused by linear methods through simple non-linear methods without using motion compensation. One successful answer is obtained through the use of median operation [14, 15]. This work uses the center-weighted-median (CWM) operation to find the correct position of the horizontally moving edge. The difference of the center weighted median filtering from the original median filtering is that the CWM uses multiple copies of the center pixel value in forming the set. For example, if seven copies of the center pixel are used along with its three left and three right neighbors, then we will need a 26-tap median filter. This filter compensates for horizontal motion with velocities up to three pixels. In addition, small details disappear in the interpolated image due to the nature of median filtering.

1.1.2 Motion-Compensated Methods

Linear methods give very good results for images where objects have very small pixel velocities. In broadcast material, however, pixel velocities usually are not very small and can even exceed 20 pixels per frame [34]. To get a very good quality interpolated image, the interpolation has to be performed along the motion trajectory. In the previous section we saw that the CWM can achieve this for very small pixel velocities. It is also attractive due to its low complexity implementation. However, the quality the CWM can offer is limited, and it can only be surpassed by motion-compensated methods.

Motion-compensated approaches in the literature can be broadly categorized into three groups based on the way they use motion estimation (ME): 1) unidirectional ME (forward or backward), 2) both forward and backward ME, and 3) bidirectional ME. Selection of the ME method influences the following steps in obtaining the interpolated image. If the first or second group is used [17,45,92], there will be overlapped areas and holes in the interpolated image; hence, interpolation has to be modified to take care of occluded regions. The use of the third group [27,62] does not result in overlapped areas and holes in the interpolated regions, i.e., each pixel is interpolated by two pixels from the neighboring frames. However, in occlusion regions it does not give very satisfactory results. This is due to the incorrect motion vector obtained in these regions. Bidirectional ME cannot give accurate results if there is covering or uncovering in the region.

These methods can further be categorized into two groups: linear and non-linear. Linear methods can be expressed similarly as in Equation 1, with a modification on $\boldsymbol{x}$ by a shift in reference frames proportional to the pixel displacement $\boldsymbol{d}$ between $I_k[\boldsymbol{x}]$ and $I_{k+1}[\boldsymbol{x}]$. Accordingly, motion-compensated frame averaging is obtained as

$$I_{k+\alpha,mca}[\boldsymbol{x}] = (1-\alpha)I_k[\mathbf{x}-\alpha\mathbf{d}] + \alpha I_{k+1}[\boldsymbol{x}+(1-\alpha)\boldsymbol{d}], \qquad 0 \leq \alpha \leq 1 \qquad (2)$$

where $\boldsymbol{d}$ indicates the displacement or motion vector. Similarly, motion-compensated picture repetition is obtained by only shifting the pixels in $I_k[\boldsymbol{x}]$ as

$$I_{k+\alpha,mcr}[\boldsymbol{x}] = I_k[\boldsymbol{x} - \alpha\mathbf{d}], \qquad 0 \leq \alpha \leq 1. \tag{3}$$

Linear motion-compensated methods decrease the artifacts, but do not eliminate them completely. To further decrease or eliminate the artifacts, non-linear methods are employed. Non-linear methods further use order statistics to improve the quality of the interpolated image. Ojo and de Haan [34, 76] utilizes median filtering to introduce static median filtering as

$$I_{k+\alpha,sta}[\boldsymbol{x}] = \mathrm{med}\big(I_k[\boldsymbol{x}], I_{k+1}[\boldsymbol{x}], I_{k+\alpha,mca}[\boldsymbol{x}]\big), \qquad 0 \leq \alpha \leq 1. \tag{4}$$

and dynamic median filtering as

$$I_{k+\alpha,dyn}[\boldsymbol{x}] = \mathrm{med}\big(I_k[\boldsymbol{x} - \alpha\mathbf{d}], I_{k+1}[\boldsymbol{x}(1-\alpha)\boldsymbol{d}], I_{k+\alpha,avg}[\boldsymbol{x}]\big), \qquad 0 \leq \alpha \leq 1. \tag{5}$$

and their various combinations for frame interpolation.

1.2 Organization of the Dissertation

The organization of the dissertation is as follows.

Chapter 2 gives a brief overview on the human visual system (HVS). Understanding of the HVS is important for understanding how we perceive images or videos.

Chapter 3 explores a new low-complexity fast motion estimation (FME) algorithm to be used in FRC. A new FME algorithm capable of producing sub-sample motion vectors at low computational-complexity is presented. Unlike existing FME algorithms, the presented algorithm considers the low-complexity sub-sample accuracy in designing the search pattern for FME. The proposed FME algorithm is designed in such a way that the block distortion measure (BDM) is modeled as a parametric surface in the vicinity of the integer-sample motion vector; this modeling enables low computational complexity sub-sample motion estimation without pixel interpolation.

Finally, the performance of the presented FME is demonstrated against the existing methods. Experimental results on video test sequences show that the proposed FME algorithm reduces computational complexity of integer- and sub- sample ME considerably compared with traditional methods at the cost of negligible performance degradation.

Chapter 4 investigates a true-motion estimation (TME) algorithm and its application to frame interpolation. A new low-complexity true-motion estimation (TME) algorithm is presented for video processing applications, such as motion-compensated temporal frame interpolation (MCTFI), or motion-compensated frame rate up-conversion (MC-FRC). Regular motion estimation (ME), which is often used in video coding, aims to find the motion vectors (MVs) to reduce the temporal redundancy, whereas TME targets to track the projected object motion as close as possible. TME is obtained by imposing implicit and/or explicit smoothness constraints on block matching algorithm (BMA). To produce better quality interpolated frames, dense motion field at the interpolation time is obtained for both forward and backward MVs; then, bidirectional motion compensation using forward and backward MVs is applied by mixing both elegantly. Finally, the performance of the proposed algorithm for MCTFI is demonstrated against recently proposed methods and a professional video production suite both objectively and subjectively. Experimental results show that the quality of the interpolated frames using the proposed method is better than the compared MC-FRC techniques.

Chapter 5 provides final conclusions about the dissertation and future research directions.

CHAPTER II

FUNDAMENTALS OF THE HUMAN VISUAL SYSTEM

2.1 Introduction

Video processing and video coding technology is advancing to better please the human observers by satisfying the requirements demanded by the human visual system. To offer better solutions that can reproduce the world we see as accurate as possible, understanding of the human visual system is crucial. In addition, a better comprehension of the Human visual system (HVS) can give rise to new insights into how video processing can be improved for artistic and scientific purposes.

Our knowledge on vision and operation of the visual system comes from various fields, such as anatomy, physiology, psychophysics, psychology, neuroscience, and molecular biology to name a few. There is immense amount of information on how our visual system works [78]; in this chapter, therefore, only a brief overview of the human visual system and motion perception will be presented to help better understand some of the requirements needed for video processing, especially for frame interpolation.

2.2 The Human Visual System

The HVS includes three main parts: eyes, brain, and the optic nerve. Eyes are analogous to a camera; the optic nerve to a transmission channel; and, the brain to a processing unit. Optical information from the eyes is transmitted to the primary visual cortex in the occipital lobe of the brain, as shown in Figure 6. Then, this information is then sent to many other visual centers of the brain. Different parts of the HVS accomplish different functions; to find out how the HVS is constructed and how it functions, let's follow the path of the light through the parts of the HVS

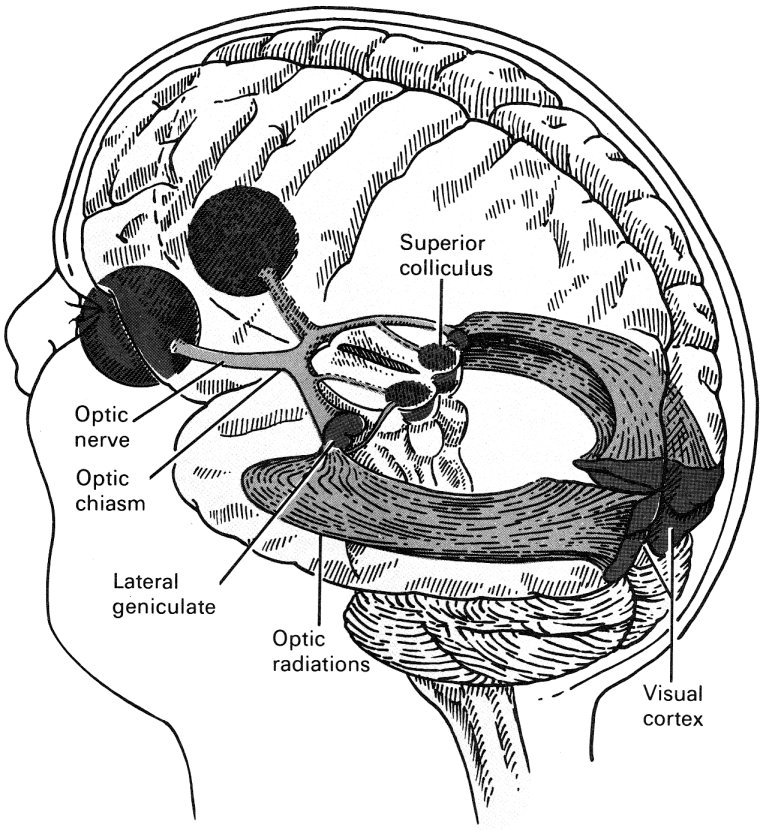


Figure 6: Human visual system. Reprinted with permission from Reference [78].

by tracing how light is converted to electrical signals, transmitted through the optic nerve to the brain, and processed in the brain.

2.2.1 The Human Eye

The analogy between a pinhole camera and eye suggested by Arabic philosopher Alhazen (A.D. 965-1040) enabled the understanding that vision occurs when light from external sources is reflected from surfaces of objects and enters the eye. Later, invention and understanding of lenses enabled Johannes Kepler (1571-1630) to study optics and further extend this to the human eye; he is considered to be the first to recognize that images are projected inverted and reversed by the eye's lens onto the retina. Since then, many advancements resulted in better understanding of the eye and visual system.

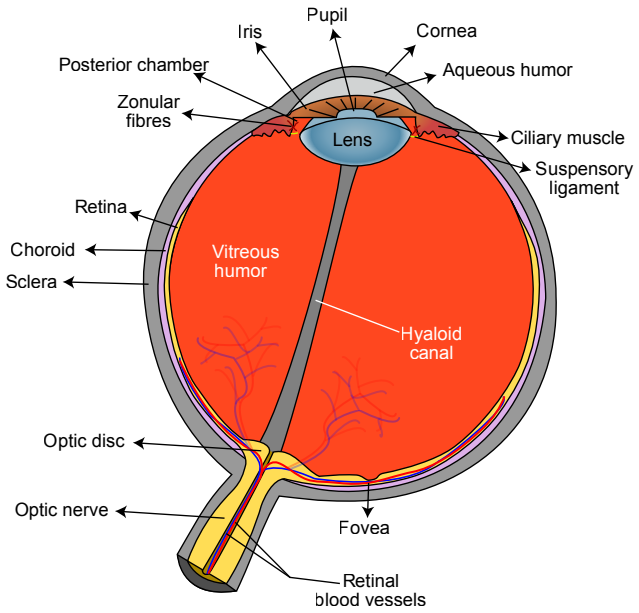


Figure 7: A cross section of the human eye.

Humans have two eyes positioned in hemispherical holes in the skull, called eye sockets, that protects them and allows them to rotate. Each one is controlled by six small but strong muscles, called extraocular muscles; these muscles are controlled by specific areas in the brain enabling scanning the visual field without turning the entire head and focusing on objects at different depths. Similar to cameras, their optical functions are gathering reflected light from the visual field and focusing it on the back of the eye as a clear image.

A cross section of the human eye is illustrated in Figure 7. What an observer perceives in a visual field is determined by processing the light emitted within or reflected from that visual field. The light is processed by the HVS by going through following parts of the eye in sequence: the cornea, the aqueous humor, the iris, the pupil, the lens, the vitreous humor, and the retina.

Light coming to the eye is first refracted at the cornea, which is a transparent protective layer with refractive surface acting as a lens. Next it passes through the

aqueous humor and enters the eye through the pupil, which is a hole in the center of the colored iris acting as an aperture stop. Then, the light is refracted again at the the lens to form a focused image on the retina. Next, the light continues through the vitreous humor, which is a transparent jellylike tissue filling the eyeball to maintain its shape. Finally it reaches to the retina. The light rays are detected by the photoreceptors in the retina and converted into neural activity. This neural activity is then communicated to the visual centers in the brain.

The amount of light reaching the retina is controlled by the iris and pupil without our knowledge. The pupil dilates or constricts depending on the illumination is low or high, respectively. In addition, both the cornea and lens bent the light so that it can be focused on the retina resulting in a sharp image. To focus on objects at different depth, lens changes its shape by adjusting its thickness using the ciliary muscles to achieve a variable focusing ability, which is called accommodation.

Although some of the light gets absorbed by lens, vitreous humor, and blood vessels before reaching the retina, majority of the light coming through the pupil gets absorbed by photoreceptors in the retina and turns into electro-chemical energy.

2.2.2 The Retina

The retina is the inner layer of the eye beneath the sclera and the choroid layers. As soon as the light reaches the retina completing its optical path, it gets converted into neural activity and continues its travel towards the brain for processing of this optical information. This energy conversion is carried out by photoreceptors located on the retina. Photoreceptors are specialized retinal cells that are stimulated by light energy through a complex process.

There are two types of photoreceptors in the retina: rods and cones. Distribution of the rods and cones across the retina is given in Figure 8. There are around 120 million rods in the retina except at the blind spot and the fovea; they are very

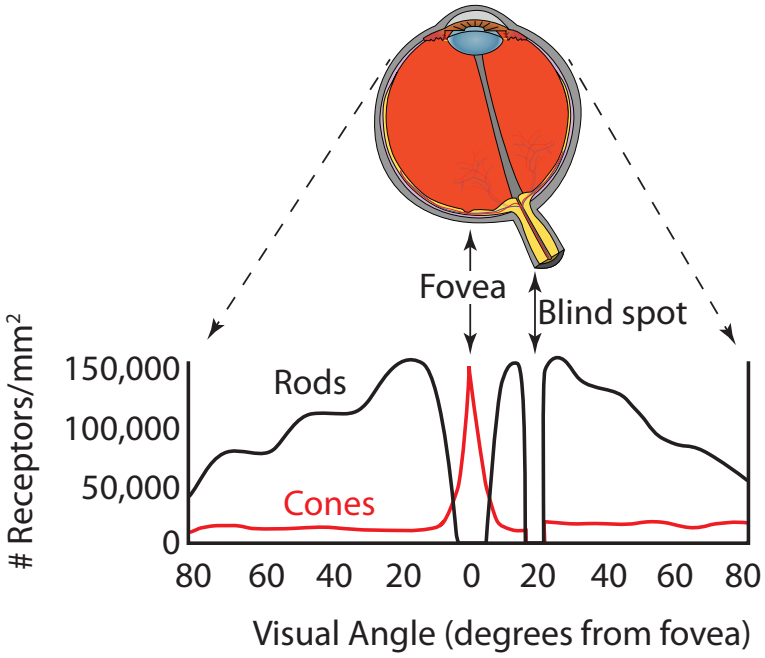


Figure 8: Distribution of rods and cones in the human retina. Adapted from Reference [78].

sensitive to light, and only used for vision at very low light levels, which is called scotopic conditions. Blind spot, which is also called optic disk, is the point where the optic nerve attaches to the eye. Fovea is a small region in the center of the retina and contains only cones. The number of cones is much less, around 8 million, and located mostly around the fovea and in lower density on the entire retina except the blind spot. In addition to color, they are in charge of our visual experiences under most normal lighting conditions, which is called photopic conditions. Although the fovea covers only about 2°, it's due to this small region we have very acute color and spatial vision. There are three classes of cones: S-, M-, and L-cones, which respond mostly to short, medium, and long wavelength regions of the visible spectrum, respectively.

As soon as the optical information is converted into neural response, some initial processing is accomplished by other types of neurons within the retina. These neurons include the horizontal, bipolar, amacrine, and ganglion cells as shown in Figure 9. The

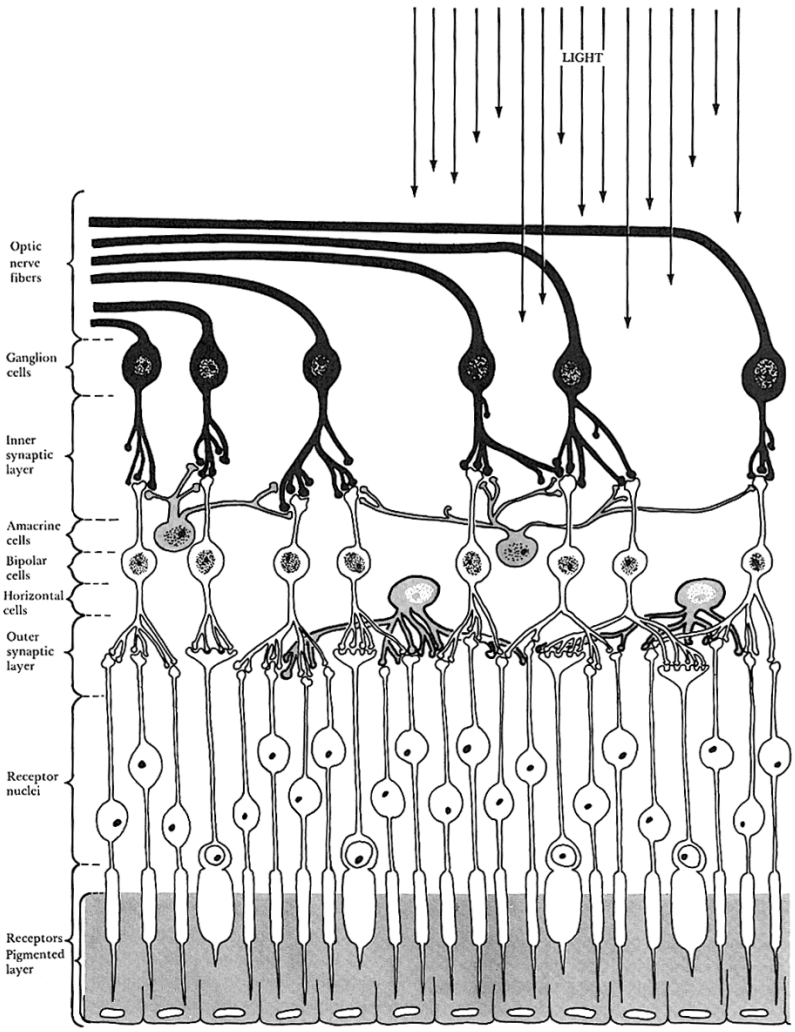


Figure 9: A cross section of retinal layers. Reprinted with permission from Reference [78].

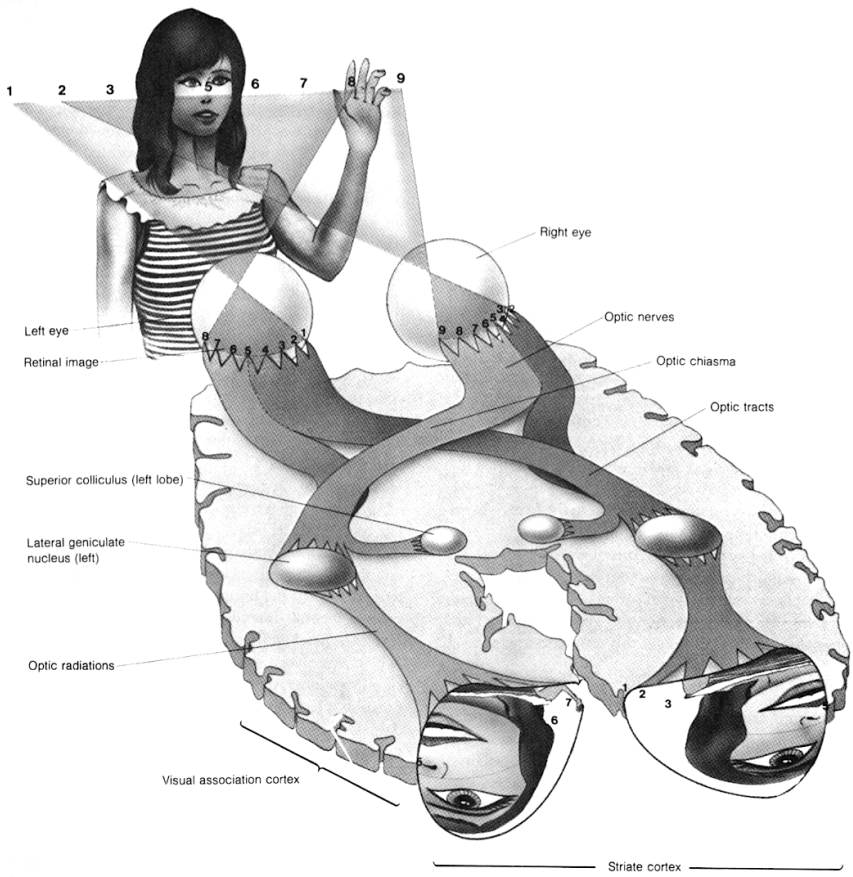


Figure 10: Neural pathway from the eye to visual cortex. Reprinted with permission from Reference [78].

neural impulse generated by photoreceptors passes through these cells and reaches to the ganglion cells. The axons of the ganglion cells come together and form the optic nerve, then leave the eye at the optic disk to the optic chiasm.

As shown in Figure 10, nerve fibers from the two eyes meet at the optic chiasm and extend to the lateral geniculate nucleus (LGN), which is a part of the thalamus, and the superior colliculus. The nerve fibers from the nasal side of the retina in each eye cross over at the optic chiasm to the opposite side of the brain, while the nerve fibers from the other side of the retina remain on the same side of the brain. From the optic chiasm, small pathway goes to the superior colliculus, which is a nucleus in the brain, and the large pathway goes first to the lateral geniculate nucleus (LGN) of the thalamus and then to the primary visual cortex, which is also called the occipital

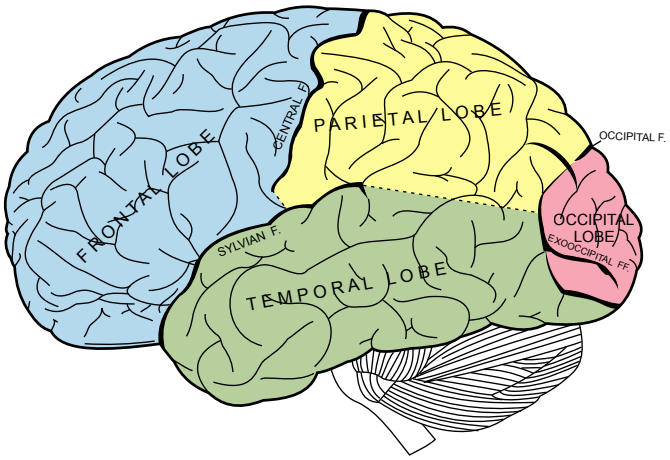


Figure 11: Four lobes of the cerebral cortex of the human brain.

cortex. Hence, each hemisphere of the brain receives visual information from the opposite side of the visual field.

Having the small portion of nerve fibers, the superior colliculus seems to process primarily information about where things are in the world and to be involved in the control of eye movements. However, the primary visual cortex processes the remaining extensive information.

2.2.3 Visual Cortex

The brain is the center of the nervous system, its main functional component is generally believed to be the neuron, which is a specialized type of cell integrating the activity of other neurons that are connected to it and propagating this integrated activity to other neurons. Brain has left and right cerebral hemispheres, which are connected by a tract of fibers known as the corpus callosum. As shown in Figure 11, there are four lobes in each cerebral hemisphere: frontal, temporal, parietal, and occipital. Occipital lobe is the primary cortical receiving area for visual information. Visual information is processed primarily in the occipital lobes, and extends to parts of the temporal and parietal lobes for more specialized processing.

The human cortex contains numerous areas that respond to visual stimuli where

processing of the various modes of vision, such as brightness, color, form, texture, depth, motion, takes place. These areas are collectively referred to as the visual cortex, which is located in the occipital lobe. The visual cortex includes the primary visual cortex, which is also known as striate cortex or V1, and extrastriate visual cortical areas, such as V2, V3, V4, and V5.

Different areas of the visual cortex are involved in different sorts of visual processing. In addition, they are organized topographically with respect to retinal locations. In fact, there is a well-established correspondence between different areas of visual cortex and different aspects of retinal stimulation, including brightness, color, motion, depth, form, and texture. Although it was initially thought that the visual information is processed serially, it is now extremely clear that a considerable amount of processing takes place in parallel across different areas.

Experiments suggest that the inferior temporal centers in the lower (ventral) system seem to be involved in recognition of objects, whereas the parietal centers in the upper (dorsal) system seem to be involved in localization, spatial orientation, and motion. These two pathways are often referred as the "what pathway" and the "where pathway", respectively.

A further possible relation between anatomical structure and physiological function has begun to emerge during the last two decades, which is called the physiological pathways hypothesis. The hypothesis is that there are separate neural pathways for processing information about different visual properties such as color, shape, depth, and motion.

Researchers traced these differences from two classes of retinal ganglion cells: one class is for color and form, the other class is for depth and motion. From the retina they project to the LGN and then to different regions of V1, V2, and MT as illustrated in Figure 12. These areas then project to distinct higher-level areas of cortex: movement and stereoscopic depth information to area V5 (also called MT, Medial

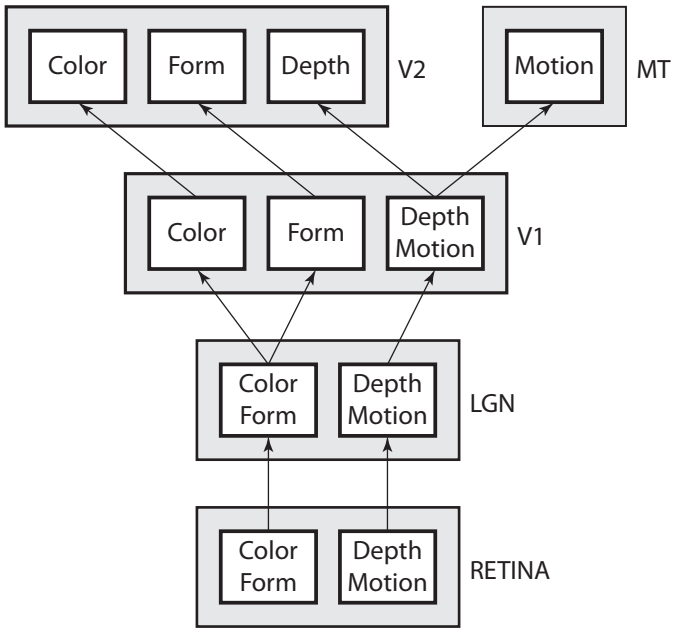


Figure 12: Schematic diagram of the visual pathways hypothesis. Adapted from Reference [78].

Temporal cortex), color to area V4, and form through several intermediate centers (including V4) to area IT (InferoTemporal cortex). From these areas, the form and color pathways may project to the ventral "what pathway" for object identification and the depth and motion pathways to the dorsal "where pathway" for object localization.

Our understanding about the visual cortex comes historically from two sources: patients with brain damage and Old World monkeys, especially macaques due to their visual ability similarities to humans. We know very little about visual processing taking place in the brain. However, recent advances in non-invasive brain-imaging techniques, such as functional Magnetic Resonance Imaging (fMRI), offer enormous promise for better understanding of the brain. Hopefully, the better understanding of the visual processing of the brain will enable us to design better technology for video displays.

2.3 Eye Movements

We have an intrinsically selective visual perception. Depending on which task we are trying to accomplish, we focus our attention selectively on certain objects in our visual field. As a result, our visual system gets more or better information on these objects. This act of visual selection can be either overt or covert. In overt visual selection, people around us can observe this, whereas in covert visual selection they cannot. For example, people around us can tell which direction we are looking at when we move our eyes, but can not tell which object or which property of this object in this direction we are focusing on. There are mainly two major functions of eye movements: fixation and tracking. Fixation refers to locating the object of interest on the fovea, where the visual acuity is highest. Tracking refers to preserving the fixated object of interest on the fovea regardless of movements of the observer or the object. Although locating the object of interest on the fovea and preserving it there can be accomplished by head and body movements as well, moving only eyes is more efficient both in terms of time and effort it requires.

The ability of the eye to move enables the HVS to examine the surroundings by selective sampling. Since the spatial and chromatic acuity are much higher at the fovea than the remaining of the retina, visual system moves the eye so that the object of interests fall sequentially on the fovea for detailed information about them. Different types of eye movements are involved when we try to gather more information about our surroundings; and, the HVS somehow stitches this spatiotemporal partially overlapping patches to create the 3-D representation of the visual field. In the following subsections we briefly examine these eye movements.

2.3.1 Physiological Nystagmus

Although we are not aware of it, our eyes are constantly making minuscule involuntary rapid movements. This is called *physiological nystagmus* and caused by tremors

in extraocular muscles. Unlike other eye movements, it's not selective. Minuscule movements cause the optical image on the retina to move slightly and continuously. Researchers found out that if the image on the retina remains on the same location, then the perception of the stabilized image mysteriously fades away within a few seconds. This finding led to many extraordinary discoveries, such as the visual system's construction of the visual field just from contour information [61].

2.3.2 Saccadic Movements

Saccades are sudden, rapid eye movements. Their function is the fixation, i.e., locating new objects of interest to the fovea. When we look around or read a book, these eye movements normally occur. A saccade is a jerky movement. When a saccade starts, its destination seems to be fixed; if this saccade misses the target, another one always follows to fixate the intended object. A single saccade takes around 150-200 ms to prepare and carry out. To process the information at different locations of an object of interest, the eye fixates on average 300 ms between saccades to process the information in each location. These sequences of fixations constitute most of our visual perception.

2.3.3 Smooth Pursuit Movements.

If there is a moving object, the eye tries to keep the retinal image of the object on the fovea by tracking it. These tracking movements are called *smooth pursuit eye movements*. Successful tracking of an object results in nearly stationary retinal image, as a result they enable the visual system to extract more spatial and chromatic information from the retinal image of the moving object.

Saccades and pursuit movements are different in various aspects:

1. **Smoothness:** Saccades are jerky and abrupt, whereas pursuit movements are smooth and continuous. However, if the motion of the tracked object is jerky, then tracking will be difficult and ,as a result, pursuit eye movements can be

jerky as well. For example, 2:3 pull-down of a panning video sequence can create such jerky visual perception at large object motions, which is usually referred as *motion judder* in the video processing literature.

2. **Feedback:** Unlike saccades, pursuit movements are not jerky; this is accomplished by employing a constant feedback correction within the visual system. Brain and eye constantly interact with each other to control the eye muscle movements so that smooth tracking can be accomplished by keeping the retinal image on the fovea. Therefore, by its very nature pursuit movements require presences of a moving object.

3. **Speed:** Saccades are much faster compared to pursuit movements; saccades can attain speeds of 900° per second, while pursuit movements can reach speeds of 100° per second.

4. **Acuity:** Although the image of the tracked object is clear in pursuit movements, objects not matching the tracking speed of the eye are perceived as smeared and unclear due to their differing motion on the retina. This blurring does not happen during saccades due to saccadic suppression.

Tracking capability of the visual system depends on the object speed; the faster the object speed becomes, the harder the tracking gets.

2.3.4 Vergence Movements

We depend on vergence movements to properly convergence or divergence the eyes to view objects at different distances from us. When we fixate and object of interest that moves toward us or away form us, vergence movements are used to track it. They are much slower than both saccades and purit movements, only 10° per second. The difference between pursuit and vergence movements is that in pursuit eye movements both eyes move in the same direction, whereas in vergence movements eyes move in

opposite directions. Both movements are employed to accurately track simultaneously if a tracked object has both depth and non-depth velocity component with respect to the observer.

2.3.5 Vestibular Movements

Saccadic, purit, and vergence eye movements are used when the head is still. When the head is moving, the eye has to compensate for that as well; in this case, vestibular and optokinetic eye movements work together to have a fixed target image on the fovea. When we rotate our head or body, vestibular eye movements keep our eyes fixated on the object of interest we have been looking at. The name vestibular is used to denote the fact that they are controlled through information supplied from the vestibular system in the inner ear, which provides information about changes in the orientation and the position of the head. Vestibular movements are faster and more accurate than pursuit movements.

2.3.6 Optokinetic Movements

When a big portion of our visual field moves uniformly across the retina, we involuntarily track the movement to the edge of our visual field, and then make a rapid movement in the opposite direction; and continue this pattern. This movement is driven by optical translations of the whole visual field so that our visual system can track the fixated object of interest in the visual field. For example, we observe this movement when we are watching outside in a moving car or train; after tracking a telephone pole through a large angle in the direction of the image motion, our eyes make a rapid movement in the opposite direction then fixate and track another telephone pole.

2.4 Characteristics of the Human Visual System

2.4.1 Spectral Sensitivity

The HVS is sensitive to a portion of the electromagnetic radiation, which is also called the visible spectrum. This band covers the wavelengths between 390 nm and 750 nm. The lens system of the eye, cornea and lens, filters the high-frequency radiation to protect the eye from potential damages by ultraviolet light.

2.4.2 Incremental Brightness Sensitivity

Experiments show that the smallest perceptible change in the luminance varies little with the value of the background luminance when it is at very low levels; the absolute change is highest around the fovea. The smallest perceptible change increases linearly with the increasing background luminance, which is known as the *Weber-Fechner* law.

2.4.3 Spatial Response

Although an imaging system satisfying the linear system theory requirements can be expressed in terms of its optical transfer function, the visual system does not fulfill these conditions. The response of the HVS is fairly homogenous near the optic axis; however, other parts are very sensitive to orientation. The response of the eye to spatial detail depends on the input pattern contrast as well. Therefore, the response of the HVS cannot be expressed with a single transfer function.

Harmonic stimuli is used for characterization of the HVS's response to contrast variations as a function of spatial frequency. Figure 13 shows the contrast sensitivity of the HVS to varying spatial frequency, this plot is called the contrast sensitivity function (CSF). The exact shape of the CSF for each individual is not necessarily the same. In facts, it depends on several factors: mean luminance, spatial position on the retina, spatial extent, orientation, temporal frequency, individual, and pathology.

The peak of the CSF is around 15 cpd and falls off rapidly on both sides. The maximum frequency is around 60 cpd.

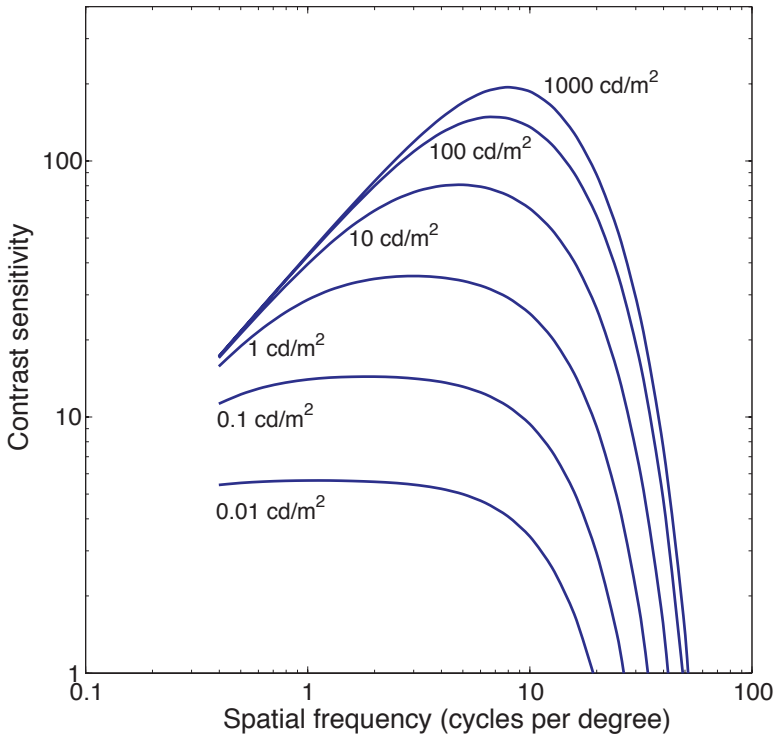


Figure 13: The CSF for different mean luminance values. Empirical formulas from Reference [8] are used to generate the CSF for each mean luminance value.

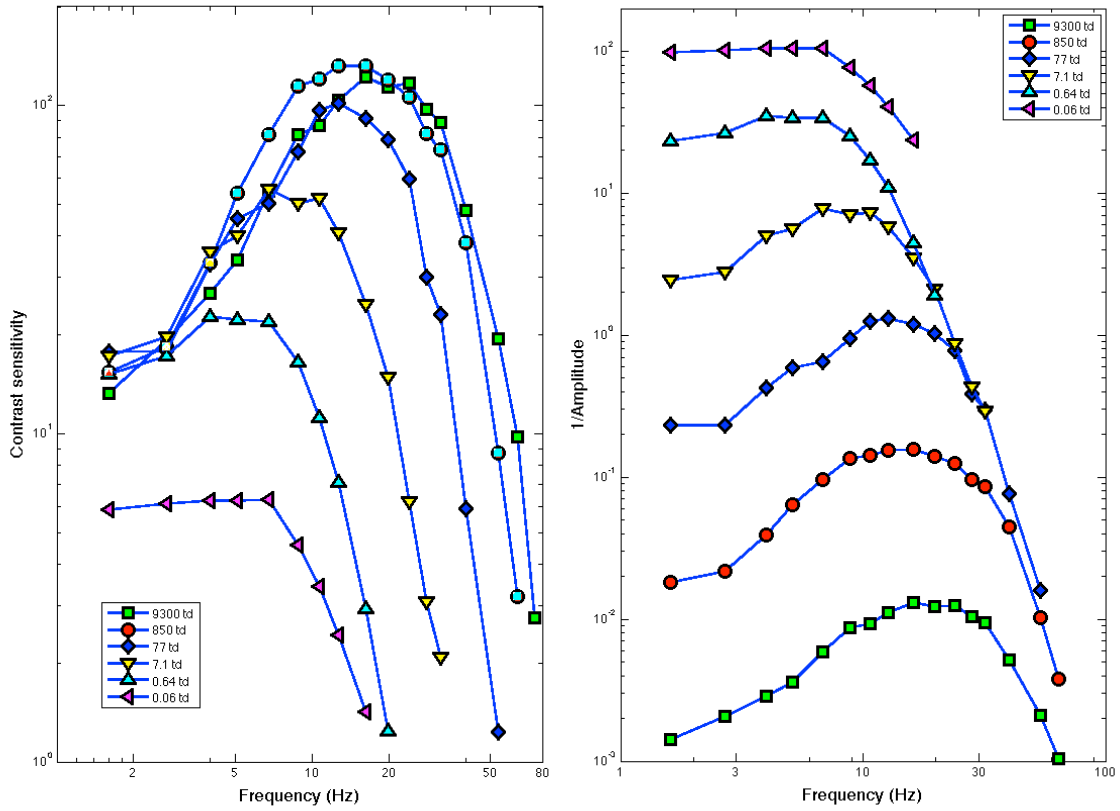
2.4.4 Temporal Response

The eye is capable of functioning at different rates of change in luminance levels similar to its ability to function over a large range of luminance levels. With the help of minuscule involuntary rapid eye movements, our eyes are continuously sampling projected retinal image repeatedly. Then, sampled information is integrated and sent to brain for further processing. Due to this repeated sampling, there is a finite amount of time for collecting, processing, and sending the data; as a result, the HVS has an inherent response limitation to rates of change in luminance levels. The ability of the HVS to see a sequence of light pulses depends on not only the luminance level but also the pulse duration that the light is on as well. If the rate is below certain threshold value, it is perceived as separate light pulses. Around this threshold value, it is perceived as a steady light with intensity changes, which is called *flicker*. Than at a certain frequency, the flicker disappears and the HVS perceives it as a steady

light; this value is called *the critical flicker frequency* or *critical fusion frequency* and is affected by several factors [59]. Critical flicker frequency is stated by Ferry-Porter law to be a linear function of the logarithm of the luminance.

The eye sums the effects of absorbed individual quanta of light over a very short period of time, which is called the *critical duration/period of vision*. Bloch's law explains this as a constant value that is equal to the product of luminance and the stimulus duration. This period of integration is different for rods and cones; it is up to 100 ms for rods and 10 to 15 ms for cones. In addition, temporal integration depends on the size of the stimulus as well; decreased stimulus size results in increased temporal integration, as a result leads to inferior temporal discrimination [59].

Similar to characterization of the spatial vision as in Figure 13, temporal vision can be characterized as well. This is called the temporal contrast sensitivity function (T-CSF), which is a plot temporal contrast sensitivity at varies frequencies. Figure 14(a) shows the measured results of the T-CSF for a typical observer for mean luminance values ranging from 0.06 to 9300 trolands at a range of frequencies; T-CSF has a bandpass characteristics with a peak around 15 to 20 Hz at higher luminance values, and with decreasing luminance value the T-CSF response becomes more like a low-pass with decreasing peak value. Figure 14(b) shows the replotted data with y-axis normalized with the absolute contrast sensitivity value, which is the contrast sensitivity value at the lowest frequency of each curve. At higher frequencies curves converge for all different luminance levels indicating that the HVS cannot adapt to the signal. At photopic light levels the CFF is around 60 Hz and at low light levels the CFF is approximately 15 Hz. Temporal resolution at low luminance values is not as efficient at that of high luminance values.



(a) (b)

Figure 14: Temporal CSF as a function of mean luminance for a large flickering field. Replotted from Reference [56].

2.5 Motion Perception

Perceiving movements is an essential part of our existence. Moving objects grasp our attention more forcefully than stationary objects. If we notice a moving object in our peripheral vision, generally we instinctively turn our eyes to better understand what is really happening as it could be an object that could harm us. In the HVS, the perception of motion of an external stimulus can be produced in one of the following ways:

- **Real:** This is the usual perception under normal conditions; an object physically is displaced from one point to another in continuous time.
- **Apparent (stroboscopic):** This is an illusion of movement that can be created by rapid presentation of completely static images.
- **Induced:** If the background surrounding an object moves one way, the object may be seen to move in the opposite way; for example, the moon can appear to move behind the clouds.
- **Autokinetic:** When a stationary spot of light is viewed for an extended period of time in total darkness then it is perceived as moving.
- **Aftereffects:** When an observer stares for an extended period of time at a field of image elements moving at a constant direction and speed, the HVS adapts to this motion. Subsequent motion perception after this motion stops will appear to move in the opposite direction. For example, staring at a waterfall can create this perception.

In this section, our focus is on the apparent motion and the other cases are beyond the scope of this dissertation.

2.5.1 Apparent Motion

Apparent motion is not a real motion, motion perception in the HVS is the result of rapid presentation of completely static images; hence, it is a visual illusion. Today's motion picture technologies, such as television, film, computer graphics, and video, rely on this illusion; sequence of pictures or frames are either captured using video camcorders or created using some other methods, and these frames are later displayed at a proper rate.

Early uses of apparent motion were by researchers in psychology and physiology. First studies on apparent motion was by Exner in mid 1870s, he used regular alternation between the two stimuli separated by a distance. At a specific rate of alternation and distance, observers always perceived clear and continuous motion of light that moved between the two stimuli. In addition, it also received significant attention from Gestalt psychologists, especially from Wetheimer; their studies using two static lights at different alternation rates showed different motion impression at different rates. As they changed the rate from very fast to very slow, they observed the following perceptions:

- **Simultaneous flicker** When the rate of alternation is faster than about 40 Hz (25 ms per presentation), motion is not perceived from one light to the other. On the contrary, two separate flickering lights appear in two different positions.

- **Phi motion** When the rate is slowed moderately, motion is perceived between the two lights without the perception of intermediate positions.

- **Beta motion** When the rate is a slowed down to around 10 Hz (100 ms per presentation), motion is perceived; a single light appears to be moving continuously back and forth between the two lights. Intermediate states of the light between the two lights are perceived as if the light were moving continuously.

- **Sequential alternation** When the rate of alternation is slowed down to about 2Hz (500 ms per presentation), motion is not perceived. Instead, the perception changes to two distinct lights alternately flashing on and off in two different positions.

Later, another Gestalt psychologist Korte, who is a student of Wertheimer, studied the dependence of apparent motion on three parameters: intensity of the two lights, distance between the two lights, and the alternation rate.

For a successful perception of apparent motion there are two factor to consider for a proper rate. First, the display rate has to be enough for a smooth continuous movement perception; second, it has to be high enough so that flicker is not perceived. For example, films are created at 24 fps, which is good enough for smooth movement perception; however, if they are projected at this rate observers perceive flicker. As a result, each frame is displayed three times using a three-blade shutter resulting in effective 72 Hz, which is fast enough for flicker-free perception. Similarly, in CRT TVs 30 fps is good enough for motion perception. However, this is to slow for CFF; hence, displaying each frame twice by interlacing the odd and even fields results in effective 60 Hz. As a result, perception is flicker-free and continuous.

2.5.1.1 The Correspondence Problem of Apparent Motion

The existence of motion perception from apparent motion suggests that the HVS perceives motion between the corresponding objects of the consecutive frames. In real motion, correspondence problem refers to finding a unique mapping between the the points or objects belonging to two simultaneous images from slightly different viewpoints, i.e., one from each eye. As a result of this activity, the brain reconstructs the 3-D view from the 2-D retinal images of left and right eyes. In apparent motion, correspondence problem is slightly different. The HVS has to figure out the correspondence between the points or objects of the successive frames from the same view

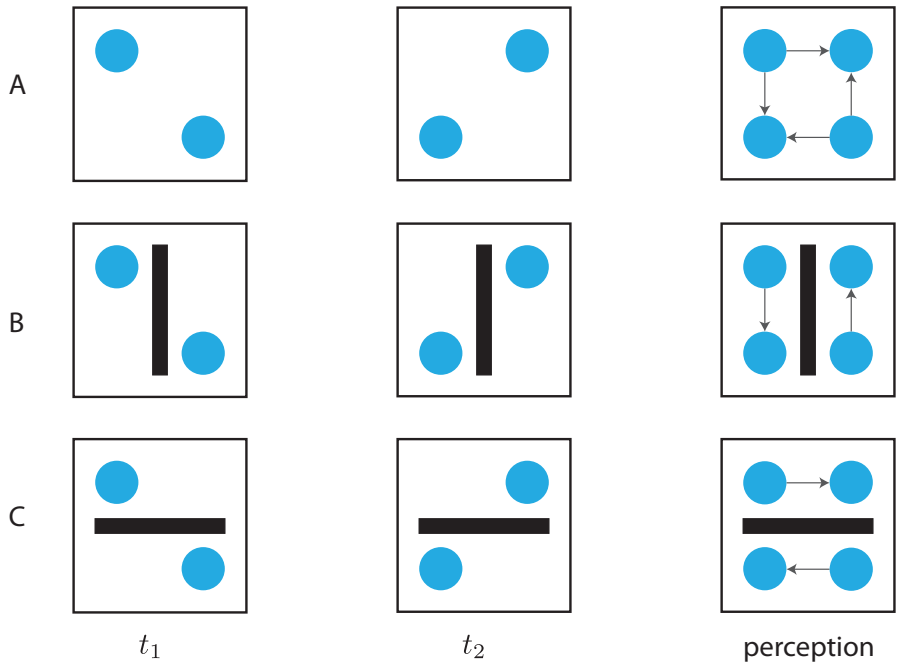


Figure 15: Correspondence of two moving dots. Adapted from Reference [78]

point.

Although apparent motion of one point using two stimuli is simple and straightforward. It may get ambiguous as the number of objects at different structures are included in the scene as it may result in many possible solutions. Figure 15 illustrates the case with only two dots. It is observed by the observers as moving horizontally, vertically, clockwise, or counterclockwise. However, when a barrier is put in between either horizontally or vertically it forces our percept to horizontal or vertical movement, respectively.

There are several factors affecting the perception; distance between the objects, inherent properties of the objects, rate of the alternation, and the neighboring objects. Currently accepted explanation states that the most forceful of the factors in solving the correspondence is the distance between the potentially corresponding objects. If all other factors are the same then the closest objects will be perceived as corresponding by the HVS. A good example for this is the wagon-wheel illusion in

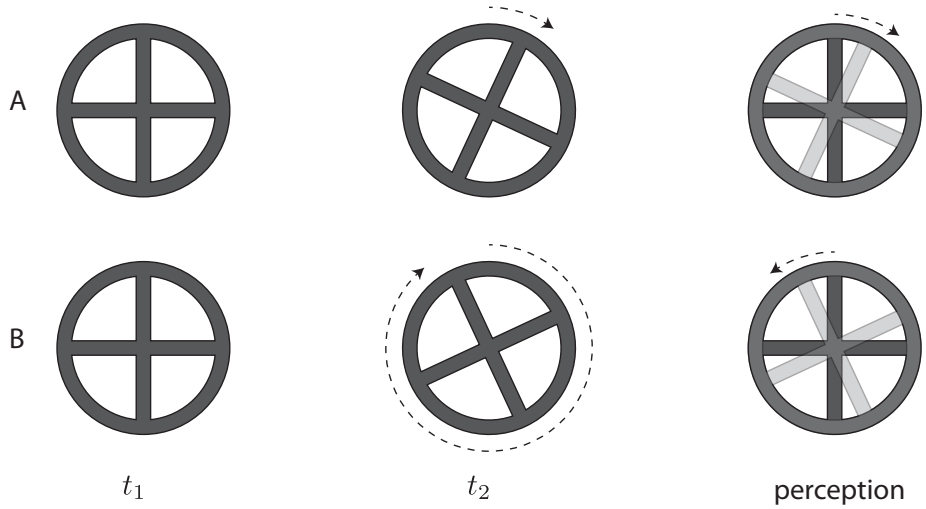


Figure 16: The wagon-wheel illusion. Adapted from Reference [78].

which spoked wheel appears to rotate backwards. A four-spoked wheel is illustrated in Figure 16. If the rotation angle per frame modulo 90° is between 0° and 45°, then the wheel appears to rotate forward; if the rotation angle per frame modulo 90° is between 45° and 90°, then the wheel appears to rotate backward.

Moreover, frame display rate can also influence the solution to the correspondence problem. For example, the motion of the three dots shown in Figure 17 is perceived differently at different rates. When the rate is slow, all three dots appear to move as a group; whereas, when the rate is fast only the outermost dot appear to move while the other two dots remains still. Recent studies suggest a trade-off between time and distance in solving the correspondence problem; at high frame rates shorter distance is preferred, whereas at slow frame rates longer distance is preferred.

There is another version of the correspondence problem called the aperture problem that emerges in both real and apparent motion. Its importance in motion perception comes from the fact that the cells coding motion in the HVS respond as if they are viewing a small portion of the visual field through an aperture due to their small receptive fields. The aperture problem means the local ambiguity in the velocity of motion due to its partial view through an aperture. For example, Figure 18 shows

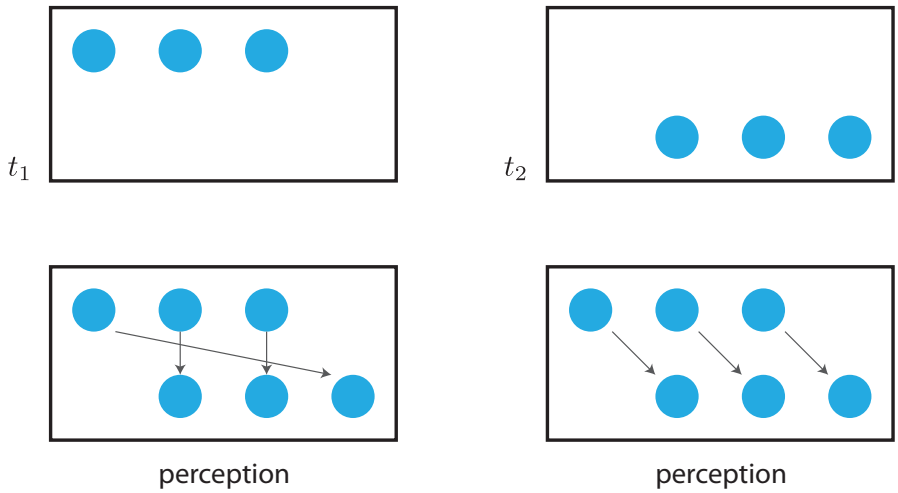


Figure 17: Correspondence of three moving dots. Adapted from Reference [78].

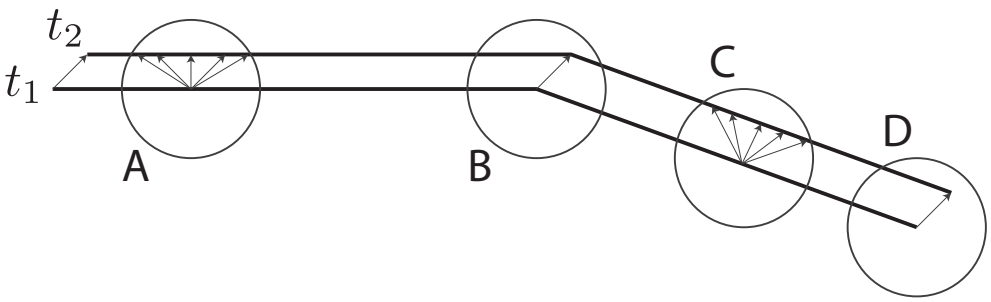


Figure 18: Different apertures of a diagonally moving line. Adapted from Reference [78].

a line moving diagonally, it shows the position of the line at times t_1 and t_2. Since the line is uniform, there are no unique points to help with the correspondence at different times at aperture problem areas. There is ambiguity in apertures A and C; however, actual motion will be perceived in apertures B and D due to existence of unique points. Although apertures A and C are ambiguous, observers usually perceive them as moving perpendicular to their orientation. This perception corresponds to the minimum speed and distance line could have.

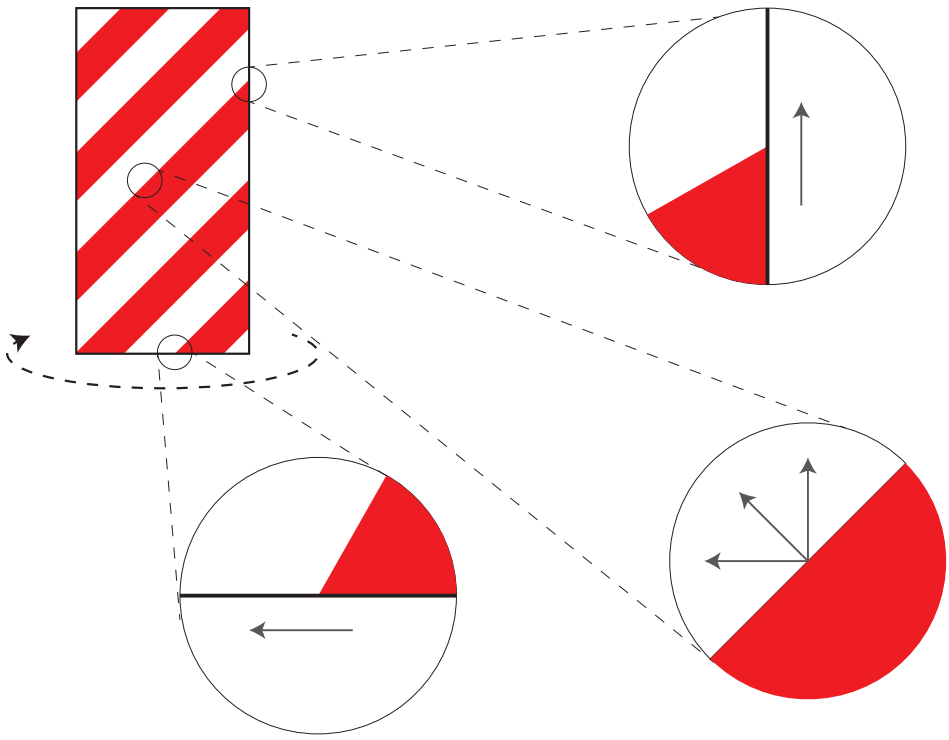


Figure 19: Illustration of the barber pole illusion. Adapted from Reference [78].

2.5.1.2 The Aperture Problem

There is a tendency in the HVS to extrapolate the unambiguous motion of unique points to other parts of the same object if there is no conflict. A good example for this tendency is the barber pole illusion, as illustrated in Figure 19. A barber pole is a cylinder painted with spiraling red and white stripes. Although it rotates continuously around its central vertical axis, it is usually perceived as moving upwards.

It is obvious that along the middle of the stripes there is ambiguity in apertures. However, at the left and right ends of stripes they appear to move upward, at the bottom and top ends of stripes they appear to move leftward. It turns out that the longer side dominates the perception as there are more stripes on the longer side, as a result the barber pole appears to move upward. If the barber pole was short and thick instead of being long and thin, then it would appear to move leftward.

CHAPTER III

FAST MOTION ESTIMATION WITH INTERPOLATION-FREE SUB-SAMPLE ACCURACY

3.1 Introduction

Motion estimation (ME) algorithms play a significant role in video coding and several video processing applications, such as noise reduction, de-interlacing, and frame rate up-conversion (FRC). Compared with other ME methods, block matching algorithms (BMAs), especially, gained widespread adoption partly due to their straightforward and simple implementation, and hardware-friendly parallelized structure; hence, BMA is part of different techniques and standards, such as ISO MPEG-1/2/4, ITU-T H.261, H.263, and H.264 [1–5]. In BMA, the current video frame is divided into macroblocks, and for each macroblock a displacement vector is found by searching the reference video frame within a predefined search area for the best matching macroblock. The resulting displacement vector is the estimated motion for the corresponding macroblock. The search process is restricted to a limited search area in the reference frame; exhaustively searching all possible candidates within this search area for the best match is called the full search (FS). During the last three decades, numerous algorithms were proposed that targeted the reduction of the computational complexity of FS with minimal performance degradation. A possible classification of these attempts results in five categories: reduction of search points (SPs), simplification of matching criterion, multi-resolution search, predictive search, and fast full search. Existing FME algorithms employ one or a combination of these categories. The focus of this paper falls in the reduction of SPs category.

Different block matching strategies and the corresponding search patterns with

various shapes and sizes have tremendous effect on both search speed and performance. These attempts started with the three-step search (TSS) and the 2D-logarithmic search (2DLOG) [49, 58] techniques. These and similar reduced step search algorithms make use of the unimodal error surface assumption, which states that the matching error monotonically decreases towards the global minimum. TSS employs rectangular patterns that decrease in size using three consecutive steps. 2DLOG employs the search in a linear direction. Due to its simplicity and effectiveness, TSS became the most popular among FME algorithms and found its way into some coding standards as well. These methods achieve significant computational complexity reduction at the cost of decreased estimation accuracy. Later, it is recognized that the unimodal error surface assumption is not always satisfied for the whole matching surface. In fact, these algorithms were easily trapped into a local minima because of the initial uniform distribution of SPs. These algorithms are very efficient for sequences with large motion vectors. However, that is not the case for sequences with small motion vectors. Li et al. [67] pointed out that the distribution of motion vectors for real-world video sequences is in fact highly center-biased. With the realization of the center-biased nature of motion vector (MV) distribution, the new three-step search (NTSS) [67] algorithm paved the way to faster ME techniques.

Several other algorithms improved on the NTSS by employing early termination, such as the four-step search (FSS) [80], the diamond search (DS) [91, 108], the cross-diamond search (CDS) [21], the hexagon search (HS) [106, 107], and the cross-diamond-hexagon search (CDHS) [22]. The NTSS modifies the first step of the TSS by including the 8-neighbors of the central point and applying early termination if the resulting MV is $(0, 0)$. If the resulting MV is one of the 8-neighbors of the central point, then the NTSS checks the remaining 8-neighbors of the first step result and then either stops or continues like TSS, depending on the minimum being one of the central points or not. NTSS performs better than TSS both in terms of minimizing

the number of search points (NSPs) and not compromising the estimation accuracy. TSS has 25 fixed SPs to check. But, the NSPs for NTSS varies between 17 and 33. FSS focuses on reducing the worst case computational requirement of NTSS at a negligible loss in estimation accuracy. In the first step, FSS uses the central point and its two-pixels away 8-neighbors[1]. In the second and third steps it continues by checking the remaining two-pixels away 8-neighbors of the SP giving the minimum estimation error if it is not the central pixel. In the fourth step, it checks one-pixel away 8-neighbors. FSS reduces the worst case computational requirement from 33 to 27 SPs and lowers the average NSPs. Its performance is between TSS and NTSS. DS, CDS, HS, EHS, CDHS, and other similar algorithms continued the same path and decreased the average NSPs even further with a small compromise in the estimation error [21, 22, 91, 106–108].

All of these algorithms perform integer-sample ME, and oftentimes sub-sample accuracy (SSA) is needed to increase the accuracy of ME. Although these FME algorithms can be easily modified to perform sub-sample ME, straightforward extension is usually obtained through a hierarchical approach and results in increased computational complexity and excessive storage requirements. This increase, however, conflicts with the objective of the FME and seriously degrades its efficiency. Performing sub-sample ME requires available sub-sample locations of the reference frame. To obtain MVs with SSA, once integer-sample ME is performed, the neighboring eight half-sample locations around the best integer-sample location are tested to find the best 1/2-sample location and this procedure is repeated for lower sub-sample levels. These sub-sample locations are obtained by interpolation of the reference frame and this needs to be done in advance; also, this process is time consuming and requires increased memory space. Matching criteria evaluated at these sub-sample locations result in sub-sample MVs. Depending on the application, 1/2-sample, 1/4-sample, or

[1] n-pixels away 8-neighbors refer to pixel locations $(\mp n, 0)$, $(0, \mp n)$, $(\mp n, n)$, and $(\mp n, -n)$

1/8-sample accuracy is used.

To the authors' knowledge, a survey in the area of FME techniques reveals that SSA with FME has not been addressed by the aforementioned algorithms. Presumably, they assume the availability of the pixel values at the sub-sample locations and continue the search hierarchically to the lower sub-sample levels, which requires $8n$ SPs to check for $1/2^n$-sample accuracy. Since the hierarchical approach of obtaining MVs with SSA results in increased complexity, FME algorithms producing MVs with SSA at lower complexity are highly desirable. Obtaining sub-sample MVs without using intensity interpolation can be achieved by modeling the estimation error as a parametric error surface in the locality of the integer-sample. Hill et al. [41] modeled the surface as a parabolic surface with six parameters and showed that this model is reasonable for any stationary two-dimensional signal and extremely close to the actual interpolation surface for sources with Gaussian-shaped autocorrelation functions. The method assumes that the block distortion measure (BDM) at the integer-sample MV point and its 8-neighbors are known. Hence, it can only be applied to the FS, TSS, FSS, and DS out of the aforementioned algorithms. Modeling the surface with fewer parameters, i.e., five, is also addressed [23, 41], where the BDMs at the location of the integer-sample MV point and its 4-neighbors are used. However, the performance obtained with this model is inferior to models using 8-neighbors. Hence, we concentrate only on models using 8-neighbors so that the performance is not lessened. It is clear that the reduction of SPs and sub-sample ME are decoupled for most of the FME algorithms in the literature. In this paper, we propose a FME algorithm that considers both SP reduction and sub-sample ME simultaneously. The proposed method can produce MVs with SSA through FME and without using intensity interpolation for sub-sample ME.

In Section 3.2, the proposed integer-sample FME algorithm is presented; in the rest of the paper this method will be referred to as the 8-neighbor search (ENS)

algorithm. Then, in Section 3.3, sub-sample accuracy is reviewed and a modification is presented. Experimental results and discussion are presented in Section 4.5. Finally, Section 4.6 concludes the paper.

3.2 Eight-Neighbor Search Algorithm

Although the ENS algorithm has no restriction on search window size similar to DS, CDS, and HS, a search window of $w = \pm 7$ with a block size of $B \times B$ is considered here for easier comparison to other FME algorithms and explanation. Also, it exploits the center-biased characteristics of MV distribution. The goal is not only to reduce the average NSPs but also to ensure that the BDM of the integer-sample MV point along with its 8-neighbors are calculated so that the subsequent interpolation-free SSA model can be used successfully. In this section, an integer-sample FME algorithm achieving this purpose is presented. For ease of explanation, n-pixels away 8-neighbors is defined as

$$\mathcal{N}_8^n = \{(x, y) | x = -n, 0, n \; ; \; y = -n, 0, n\} \setminus \{(0, 0)\}, \tag{6}$$

where (x, y) denotes the rectangular coordinates relative to the location of the pixel of interest. Clearly, $\mathcal{N}_8^n$ for $n = 1$ gives the immediate 8-neighbors of the pixel of interest.

The proposed method utilizes the set of points $\mathcal{N}_8^1 \cup \{(0, 0)\}$ in the first step similar to the NTSS algorithm. The details of the ENS algorithm for a search window of $w = \pm 7$ are as follows:

Step 1) Points in $\mathcal{N}_8^1 \cup \{(0, 0)\}$ of the search center are checked, and the minimum BDM point is found. Then, the remaining immediate 8-neighbors of the minimum BDM point are checked. If the minimum BDM point is the search center, the search is stopped, i.e., MV=$(0, 0)$. Otherwise, go to Step 2.

Step 2) Points in $\mathcal{N}_8^3$ of the search center are checked. If the minimum BDM point

is the search center, the search is stopped, i.e., MV=$(0,0)$. Otherwise, go to Step 3.

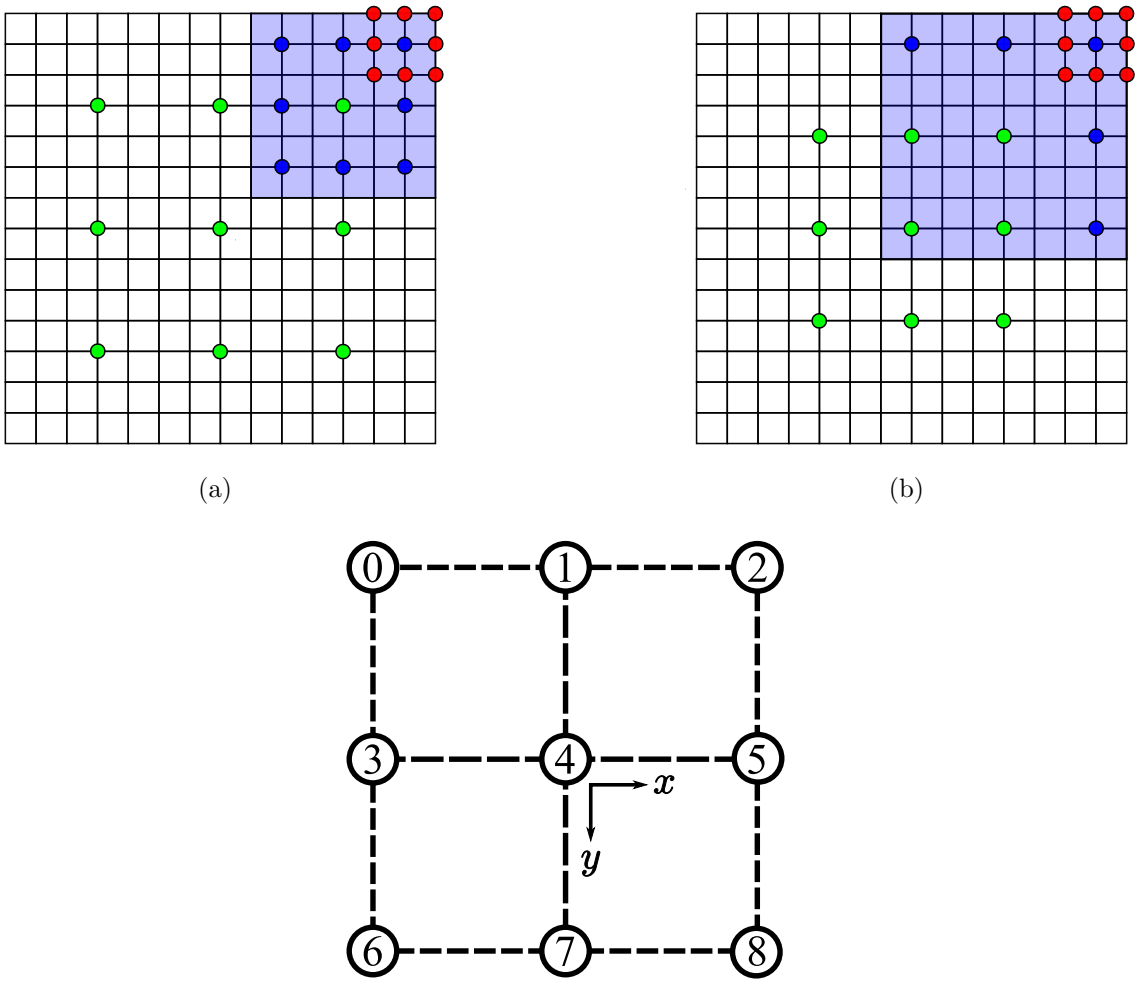
Step 3) Check the remaining 8-neighbors $\mathcal{N}_8^3$ of the minimum BDM point. Update the minimum BDM point. Check the immediate 8-neighbors $\mathcal{N}_8^1$ of the minimum BDM point.

The point giving the minimum BDM is declared as the integer-sample MV for this block.

The proposed method selects the step sizes more prudently compared with TSS, NTSS, and similar algorithms. They check the four-pixels away 8-neighbors, $\mathcal{N}_8^4$, first. Then, the two-pixels away 8-neighbors, $\mathcal{N}_8^2$, of the minimum BDM point are checked. Finally, the immediate 8-neighbors, $\mathcal{N}_8^1$, of the minimum BDM point are checked. Hence, the step sizes of these algorithms are $4-2-1$, whereas, the proposed algorithm uses $3-3-1$, instead. The approach taken by these methods decreases the size of the possible reachable area faster than the proposed method. As can be observed from Figure 20, the number of possible reachable points by TSS at each step are $15^2 \rightarrow 7^2 \rightarrow 3^2$, whereas for the proposed algorithm they are $15^2 \rightarrow 9^2 \rightarrow 3^2$; the reachable points at the second step are shown with a light-blue shade in the figure. Compared with TSS, the first step size gives a more center-biased nature and the second step size offers a larger reachable area to the proposed algorithm.

3.3 Sub-sample Accuracy without Interpolation

In applications employing ME, oftentimes integer-sample MV is not sufficient to achieve the desired quality level; hence, integer-sample ME is followed by sub-sample ME for improvement. Sub-sample accuracy improves the quality substantially at the cost of increased complexity; sub-sample ME constitutes a significant amount of the overall computational complexity [102]. As a result, a coarse-to-fine manner is used to refine the accuracy of the MV. First, the coarse MV is found by integer-sample



(a) (b) (c)

Figure 20: a) $4-2-1$ step-size illustration, b) $3-3-1$ step-size illustration, c) illustration of the integer-sample MV point and its 8-neighbors. In illustrations a) and b), the light-blue shaded area indicates the potential reachable search points in the second stage.

ME. Then, refinement can be achieved by two different approaches: 1) direct interpolation of the reference frame followed by sub-sample ME or 2) 2-D polynomial surface modeling of the BDM in the vicinity of the integer-sample MV.

For the first method, refinement is usually achieved as follows: first, interpolate the reference frame search area at sub-sample locations, and then perform sub-sample ME around the coarse MV. Evidently, there will be repeated interpolation of some of the pixel neighborhoods since the search area of different blocks may overlap. One alternative to this approach is to pre-compute the sub-sample location data for the entire frame; however, this results in increased memory requirements. Using short-length interpolating filters, and employing sub-sample ME in a hierarchical fashion in the vicinity of the selected integer-sample MV are among the common techniques used to decrease the complexity of the sub-sample ME. Regardless of how the sub-sample data is obtained, additional $8n$ SPs are checked for $1/2^n$-sample accuracy in hierarchical sub-sample ME approach. Hence, real-time application use of this method becomes harder to justify because of its increased computational complexity. Also, the need for more accurate estimation of MVs is increasing. For example, in recent video compression standards, as low as 1/8-sample accuracy is employed to achieve a better RD optimization.

For the second method, refinement is achieved by modeling the BDM values in the vicinity of the integer-sample MV as a 2-D polynomial surface. There are different estimation models and methods. Estimation models range from using integer-sample MV point along with its 4-neighbors to 8-neighbors, which gives rise to 2-D polynomial surface models having a different number of parameters. Solutions of these models result in different solution methods ranging from underdetermined models to overdetermined models. Hill et al. [41] discuss a model that is controlled by six parameters, which is called complete-system model (CSM); after pointing out the inferiority of the underdetermined model that uses 4-neighbors and the overdetermined

model that uses 8-neighbors, they propose a model that uses 4-neighbors and one of the diagonal neighbors such that this diagonal neighbor point best fits the model compared with other diagonal neighbors. Hence, a total of five neighbors along with the integer-sample MV point is used to estimate six parameters of the 2-D parabolic surface. In the next subsection, we give the details of this model.

3.3.1 Parabolic Model

Hill et al. [41] use a parametrically controlled parabolic surface model that was originally suggested by Giunta et. al [38] as

$$f_d(x, y) \approx f(x, y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F, \tag{7}$$

where f is the estimated BDM value of a block, and x and y are the coordinates of the estimation centered at the integer-sample MV of that block. Both x and y vary between $[-1.0, 1.0]$. Hence, after obtaining the unknown coefficients of the model from data, estimated BDM values at sub-sample locations can be easily calculated using $f(x, y)$, and the location giving the minimum value,

$$\hat{\mathbf{x}} = (\hat{x}, \hat{y}) = \underset{x,y \in (-1,1)}{\arg\min}\ f(x, y), \tag{8}$$

can be found using a heuristic or gradient search type algorithm.

Once the integer-sample ME is finished, BDM values of the integer-sample MV point and its 8-neighbors, which are shown in Figure 20(c), are available. Approximated BDM value, (7), evaluated at these nine points gives nine equations. Then, these equations are used to estimate the unknown parameters $A, B, C, D, E,$ and F.

The proposed solution of Hill et al. [41] is

$$\begin{aligned} A &= -f_4 + \frac{1}{2}(f_3 + f_5) \\ B &= \underset{B_k \forall k \in \{0,2,6,8\}}{\arg\min} \sum_{i=0,2,6,8} |f_i - f_{d,k}| \\ C &= -f_4 + \frac{1}{2}(f_1 + f_7) \\ D &= \frac{1}{2}(f_5 - f_3) \\ E &= \frac{1}{2}(f_7 - f_1) \\ F &= f_4, \end{aligned} \tag{9}$$

where f_j, for notational simplicity, denotes the value of the BDM evaluated at the pixel location labeled with $j = 0, 1, ..8$ and $f_{d,k}$ is the estimate of f_k using (7) with parameters $A, B_k, C, D, E,$ and F. B_k is the value of B found with the system of equations using points 1,3,4,5,7, and k. Hence, CSM uses the inteegr-sample MV point, its 4-neighbors, and one of the diagonal neighbors. Although CSM gives better results compared with overdetermined and underdetermined models, they are clearly discarding three data points.

It is known by the Stone-Weierstrass theorem [86, 87] that any 2-D continuous function defined on a closed interval can be uniformly approximated by a polynomial with two variables. In addition, sub-sample maps at 1/8-accuracy presented by Chiew et al. [23] suggest that the quality of the approximation can be increased. Hence, to better employ these three discarded data points, another model with more parameters can be used.

The model proposed in (7) can be seen as a second-order bivariate Taylor polynomial approximation as well, which can be rewritten as

$$f(x, y) = \sum_{j=0}^{2} \sum_{i=0}^{j} p_{j-i,i} x^{j-i} y^i. \tag{10}$$

Note that the number of monomials in an n-th order Taylor approximation is $r = (n+1)(n+2)/2$. The order can be increased to three to offer more flexibility, as the

second derivative will not necessarily evaluate to zero, hence allowing variation across the surface. Increasing the order to three improves the accuracy of the approximation by adding the monomials $\{x^3, x^2y, xy^2, y^3\}$. In the next subsection, the proposed method to obtain the coefficients of this bivariate polynomial is presented.

3.3.2 Parameter Estimation

The proposed polynomial approximation has ten unknown parameters, whereas only nine data points are available. To better explain the monomial selection for the given data points let's express the value of the polynomial given in (10) as

$$f(x, y) = \mathbf{m}^T \mathbf{p}, \tag{11}$$

where $\mathbf{m}$ is the monomial vector described as

$$\mathbf{m} = [m_0, m_1, \ldots, m_{r-1}]^T =$$
$$[x^0y^0, x^1y^0, x^0y^1, \ldots, x^dy^0, x^{d-1}y^1, \ldots, x^1y^{d-1}, x^0y^d]^T,$$

and $\mathbf{p}$ is the parameter vector defined as

$$\mathbf{p} = [p_{0,0}\,, p_{1,0}\,, p_{0,1}\,, \cdots\,, p_{d,0}\,, p_{d-1,1}\,, \cdots\,, p_{1,d-1}\,, p_{0,d}\,]^T,$$

and d is the order of Taylor approximation.

Evaluating (11) at nine points gives the following system of equations:

$$\mathbf{f} = [\mathbf{m}_0|\mathbf{m}_1|\cdots|\mathbf{m}_8]^T\mathbf{p} = \mathbf{M}\mathbf{p} \tag{12}$$

where $\mathbf{m}_j$ refers to the monomial vector evaluated at point j. $\mathbf{M}$ can be viewed as a matrix of columns as well, where each column corresponds to a monomial evaluated at nine points. To have a unique solution, the columns of $\mathbf{M}$ have to be linearly independent. It is observed that for given sampled values, columns of $\mathbf{M}$ corresponding to monomials $\{x^3, y^3\}$ are the same as the columns of $\mathbf{M}$ corresponding to monomials $\{x, y\}$, respectively. To have a nonsingular $\mathbf{M}^T\mathbf{M}$ matrix, we need to remove monomials $\{x^3, y^3\}$ and add one more monomial from the additional monomials of the

fourth-order approximation terms, $\{x^4, x^3y, x^2y^2, xy^3, y^4\}$. It can be easily observed that only the x^2y^2 term gives a unique solution since the other columns corresponding to terms $\{x^4, x^3y, xy^3, y^4\}$ are the same as columns corresponding to the terms $\{x^2, xy, xy, y^4\}$, respectively. Although monomials are linearly independent, for the given nine data points some monomials give the same values at these points, e.g., x and x^3, x^2 and x^4, xy and x^3y. Hence, the monomials $\{1, x, y, x^2, xy, y^2, x^2y, xy^2, x^2y^2\}$ are used to have a unique solution using the specified nine data points. In this case, the solution is

$$\begin{aligned}
p_{2,2} &= f_4 - \frac{1}{2}(f_3 + f_5 + f_1 + f_7) + \frac{1}{4}(f_2 + f_6 + f_0 + f_8) \\
p_{1,2} &= \frac{1}{2}(f_3 - f_5) + \frac{1}{4}(f_2 - f_6) + \frac{1}{4}(f_8 - f_0) \\
p_{2,1} &= \frac{1}{4}(f_6 - f_2) + \frac{1}{2}(f_1 - f_7) + \frac{1}{4}(f_8 - f_0) \\
p_{0,2} &= -f_4 + \frac{1}{2}(f_1 + f_7) \\
p_{1,1} &= \frac{1}{4}(f_8 + f_0) - \frac{1}{4}(f_6 + f_2) \\
p_{2,0} &= -f_4 + \frac{1}{2}(f_3 + f_5) \\
p_{0,1} &= \frac{1}{2}(f_7 - f_1) \\
p_{1,0} &= \frac{1}{2}(f_5 - f_3) \\
p_{0,0} &= f_4,
\end{aligned} \tag{13}$$

Comparing (9) and (13) reveals that the proposed polynomial approximation gives the same answer as CSM method for lower-order coefficients $p_{2,0}, p_{0,2}, p_{1,0}, p_{0,1},$ and $p_{0,0}$. Since all nine data points are used, the $p_{1,1}$ term is calculated directly. Besides, the accuracy is increased due to the additional terms $p_{2,1}, p_{1,2}, p_{2,2}$. Unlike CSM method, the quality of the approximation does not need to be checked. Compared with the solution (9), the solution (13) does not require minimization of a cost function to select $p_{1,1}$ term, as a result saves 12 function evaluation. Also, it is easily obtained by addition and bit shift operations.

3.4 Experimental Results and Discussion

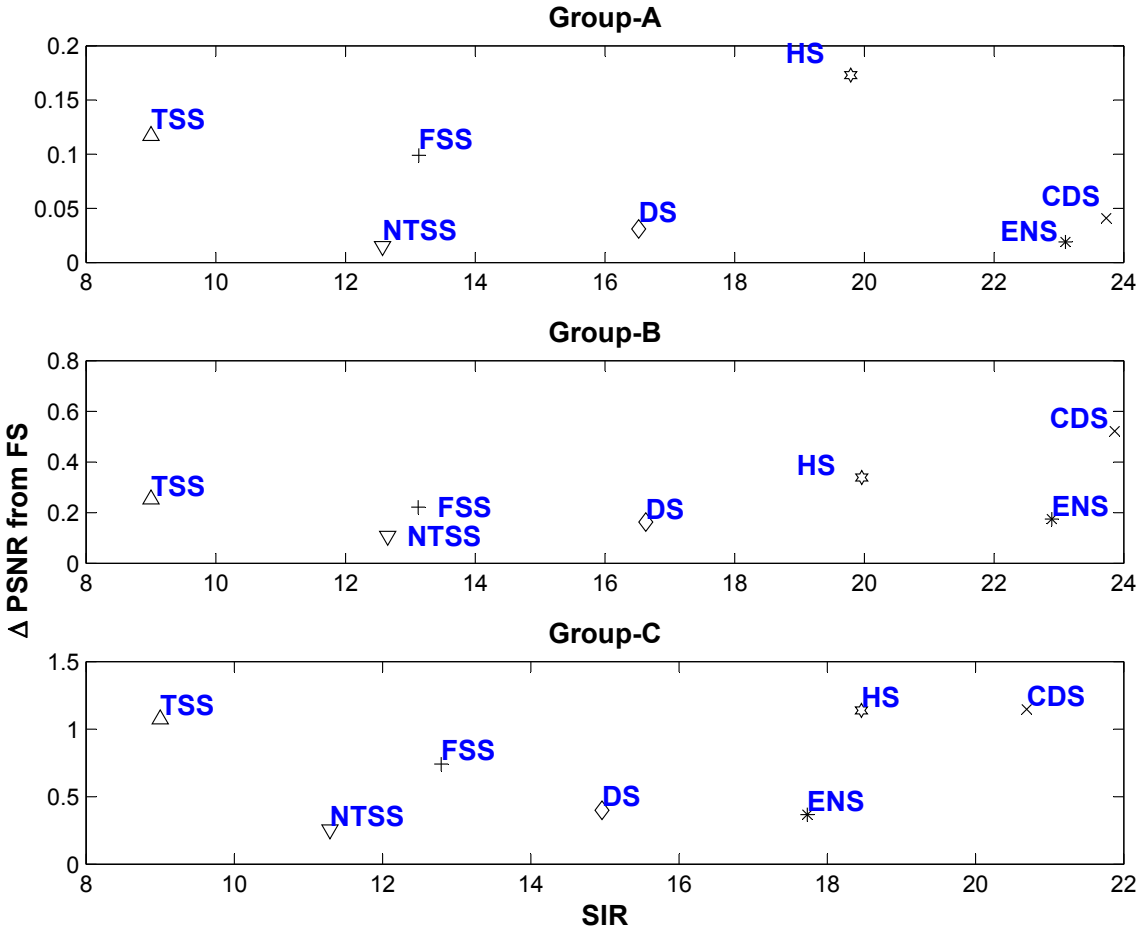


Figure 21: PSNR degradation from the FS versus SIR with respect to the FS for different FME algorithms using video sequences in Groups A, B, and C.

The proposed algorithm has been successfully tested on a variety of video sequences to evaluate its performance compared with existing similar algorithms. A few of the results are shown in this paper. To analyze only the effect of the search algorithm, unlike ME in video coding applications, backward ME is performed between successive original frames and MSE is calculated using the original and its corresponding motion-compensated frame. This approach avoids the influence of rate-distortion optimization and error propagation. ME is conducted by using only the 8-bit luminance component and sum of absolute differences (SADs) is used as the BDM.

Three different experiments have been considered. In the first experiment, the first 100 frames of 15 QCIF (176×144) sequences are tested with different FME algorithms against the FS, where a search window of $w = \pm 7$ with block size of 8×8 is used in the the simulations. These sequences are classified into three groups based on the amount of spatial detail and motion activity for easier interpretation.

- Group A: Akiyo, Claire, Container, Grandma, and Miss America
- Group B: Hall Objects, Mother & Daughter, News, Salesman, and Sign
- Group C: Carphone, Coastguard, Foreman, Suzie, and Tennis.

Although a search window of $w = \pm 7$ suffices for QCIF sequences, larger search range is required for larger resolutions. Usually, a combination of the reduction of SP technique with predictive search or multi-resolution search technique is used to achieve this. In the second experiment, four 4CIF (704×576) sequences are tested with different FME algorithms against the FS where search center prediction is used to precede the search algorithm to cover a larger search range. For search center prediction, MVs of the left, top, top-right spatial blocks; co-located previous temporal block; and $(0, 0)$ are employed. The tested sequences are City (600 frames), Harbour (600 frames), Ice (480 frames) and Soccer (600 frames). As a benchmark, the FS scheme that uses a search window of $w = \pm 32$ without search center prediction is also included.

In the third experiment, SSA is examined. To analyze only the effect of SSA method, integer-sample FS is followed by different SSA methods using the CIF sequences used in the first experiment. Bilinear interpolation is used for obtaining the sub-sample pixel values. The methods used for comparison are: full quarter-sample accuracy (FQSA), hierarchical quarter-sample accuracy (HQSA), hierarchical half-sample accuracy (HHSA), CSM method by Hill et al. [41], the proposed parametric sub-sample accuracy (PSSA), and hybrid parametric sub-sample accuracy (HPSSA).

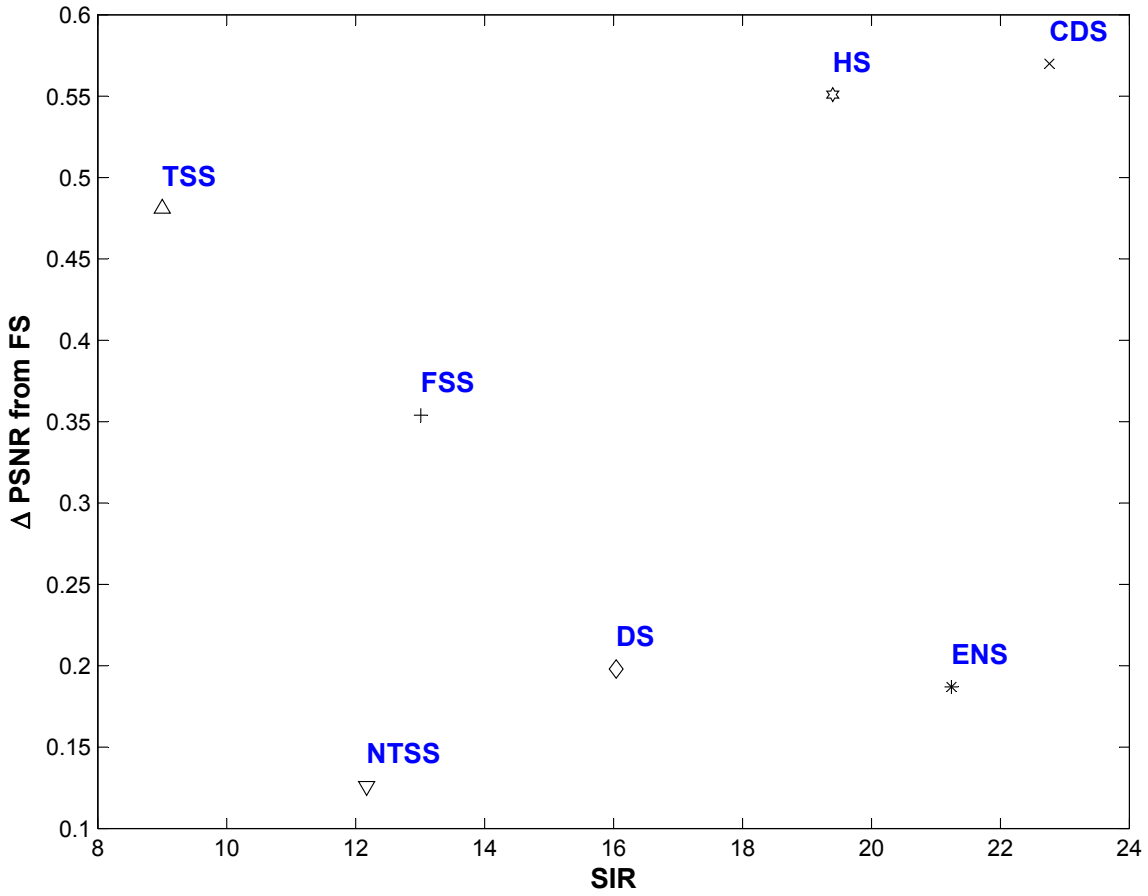


Figure 22: The average PSNR degradation from the FS versus the average SIR with respect to the FS for different FME algorithms using QCIF video sequences.

FQSA checks all available quarter-sample SPs, totaling 49 NSPs. HQSA checks 18 NSPs, i.e., first, 8 half-sample points around the best ISA point, and then 8 quarter-sample points around the best half-sample point. HHSA checks only 8 half-sample points around the best ISA point. For CSM and PSSA methods, polynomial model is evaluated at half-sample and quarter-sample points in a hierarchical manner. In HPSSA, evaluation of PSSA at half-sample points is followed by checking the SAD of immediate quarter-sample 4-neighbors.

To compare the effectiveness of the proposed ENS algorithm, both the average MSE per pixel and the average NSPs per block are used against seven other BMAs: FS, TSS, NTSS, FSS, DS, HS, and CDS. Results of the first and second experiments

are given in Table 1 and Table 2, respectively. P+ denotes the preceding search center prediction in Table 2. To make the comparison much easier, the average MSE and NSP are given for the FS; the average MSE change from the FS and the speed improvement ratio (SIR) with respect to the FS are given for the FME algorithms. In addition, the average PSNR change from the FS versus the SIR with respect to the FS are plotted for the first experiment in Figures 21 and 22.

For video sequences in groups A, B, and C NTSS gives the lowest MSE degradation from the FS and CDS gives the highest SIR. However, MSE degradation of CDS can be quite large compared with others. ENS gives very small degradation from NTSS and its SIR gets close to the SIR of CDS for CIF sequences.

As shown in the overall comparison in Figure 22, CDS is always the fastest algorithm; however, its PSNR degradation from the FS can be quite large compared with other algorithms. On the other hand, NTSS always gives the minimum PSNR degradation from the FS, but it is not as fast. CDS and HS were proposed to improve the speed of DS at the expense of degradation in MSE or PSNR quality [21, 22, 106, 107]; however, ENS achieves similar SIR as CDS and HS without compromising the MSE compared with DS. Thus, as can be seen from Table 1 and Table 2, ENS achieves the goal of the FME algorithms; it achieves a SIR close to the SIR of CDS while having a small MSE degradation from the FS. When compatibility with the sub-sample accuracy model is considered, ENS gives the best performance compared with the FS, TSS, FSS, and DS. Hence, it becomes evident that ENS is the best choice for sub-sample model that achieves the fastest performance with minimal MSE degradation from the FS.

Results of the third experiment are shown in Table 3. It shows the average improvement that can be achieved when SSA is employed. The second column shows the NSPs SAD has to be calculated. FS+ denotes the integer-sample FS that precedes the corresponding SSA. HQSA achieves as good as FQSA. As HHSA demonstrates,

around 60% of the available SSA improvement is gained by checking the 8 half-sample points, and the remaining 40% comes from checking the 8 quarter-sample points. Although the proposed model does not achieve as good as FQSA, it performs 8% better than CSM and close to HHSA while its complexity is lower than both CSM and HHSA. Quality of the PSSA can be improved through a hybrid scheme, HPSSA, from around 60% to 85% with only additional 4 SPs.

3.5 Conclusion

To reduce the computational complexity, we have proposed a FME algorithm capable of producing MVs with SSA. The proposed FME algorithm considers both SP reduction and interpolation-free sub-sample ME simultaneously. Experimental results show that the proposed algorithm substantially reduces computational complexity at the cost of negligible MSE degradation from the FS.

Table 1: Comparison of the proposed algorithm with existing algorithms using QCIF video sequences. ΔMSE denotes the MSE change from the FS, SIR denotes the speed improvement ratio with respect to the FS, and NSP denotes the number of search points.

	FS		TSS		NTSS		FSS		DS		HS		CDS		ENS	
	MSE	NSP	ΔMSE	SIR	ΔMSE	SIR	ΔMSE	SIR	ΔMSE	SIR	ΔMSE	SIR	ΔMSE	SIR	ΔMSE	SIR
Akiyo	3.00	225	0.12	9.0	0.00	13.2	0.09	13.2	0.00	17.3	0.26	20.4	0.02	24.9	0.00	24.9
Claire	3.59	225	0.08	9.0	0.01	12.8	0.06	13.2	0.03	16.6	0.22	19.8	0.04	23.9	0.01	24.0
Container	2.80	225	0.00	9.0	0.00	13.0	0.00	13.2	0.00	17.1	0.00	20.3	0.00	24.7	0.00	24.5
Grandma	3.70	225	0.08	9.0	0.01	12.3	0.07	13.0	0.02	16.3	0.17	19.6	0.03	23.5	0.02	22.4
MissAmerica	4.98	225	0.52	9.0	0.05	11.4	0.41	12.9	0.13	15.3	0.57	18.9	0.15	21.6	0.06	19.8
Average	3.61	225	0.16	9.0	0.01	12.6	0.13	13.1	0.04	16.5	0.24	19.8	0.05	23.7	0.02	23.1
HallObjects	15.82	225	0.76	9.0	0.21	12.7	0.59	13.2	0.27	16.6	0.93	20.1	0.84	23.9	0.37	22.8
MthrDotr	12.08	225	1.02	9.0	0.38	12.1	0.99	12.9	0.87	16.0	1.47	19.3	3.57	22.9	0.61	21.4
News	17.62	225	1.21	9.0	0.65	13.0	1.82	13.2	1.31	17.0	1.96	20.2	4.67	24.5	1.02	24.1
Salesman	6.53	225	0.44	9.0	0.10	13.1	0.27	13.2	0.16	17.2	0.59	20.3	0.37	24.6	0.16	24.3
Sign	25.01	225	2.71	9.0	1.80	12.4	3.26	13.1	2.75	16.4	4.01	19.7	14.16	23.4	3.36	21.9
Average	15.41	225	1.23	9.0	0.63	12.6	1.38	13.1	1.07	16.6	1.79	20.0	4.72	23.9	1.11	22.9
Carphone	20.96	225	0.44	9.0	0.10	11.2	0.27	13.0	0.16	14.9	0.59	18.6	0.37	21.2	0.16	18.1
Coastguard	41.60	225	10.74	9.0	2.31	11.2	7.96	13.0	4.38	15.4	9.20	18.8	22.88	21.0	4.43	17.5
Foreman	24.83	225	8.08	9.0	1.88	11.5	5.10	12.6	3.74	15.0	12.56	18.7	6.37	21.1	2.28	18.5
Suzie	14.51	225	3.02	9.0	0.89	11.6	2.58	12.8	1.54	15.3	4.19	18.8	15.73	21.8	1.63	18.5
Tennis	62.01	225	45.24	9.0	6.95	11.0	22.66	12.5	8.18	14.3	34.12	17.3	33.20	18.4	9.88	16.1
Average	26.36	225	13.50	9.0	2.43	11.3	7.71	12.8	3.60	15.0	12.13	18.5	15.71	20.7	3.68	17.7

Table 2: Comparison of the proposed algorithm with existing algorithms using 4CIF video sequences. ΔMSE denotes the MSE change from the FS, SIR denotes the speed improvement ratio with respect to the FS, and NSP denotes the number of search points.

	FS		P+FS		P+TSS		P+NTSS		P+FSS		P+DS		P+HS		P+CDS		P+ENS	
	MSE	NSP	MSE	NSP	ΔMSE	SIR	ΔMSE	SIR	ΔMSE	SIR	ΔMSE	SIR	ΔMSE	SIR	ΔMSE	SIR	ΔMSE	SIR
City	32.73	4225	35.44	225	1.06	9.0	0.87	12.2	4.65	13.2	4.15	16.5	4.90	20.1	5.12	23.4	4.43	22.0
Harbour	41.98	4225	41.89	225	1.01	9.0	0.71	12.3	1.42	13.2	0.81	16.3	2.03	20.2	1.83	23.4	1.01	21.5
Ice	12.71	4225	10.85	225	1.10	9.0	0.91	12.0	1.44	13.1	1.09	16.1	1.64	19.5	3.55	22.9	1.27	21.4
Soccer	81.03	4225	28.36	225	2.73	9.0	2.43	11.6	4.65	13.0	3.53	15.5	4.38	19.0	16.79	22.1	4.07	19.2
Average	42.11	4225	29.14	225	1.47	9.0	1.23	12.0	3.04	13.1	2.39	16.1	3.24	19.7	6.82	23.0	2.69	21.0

Table 3: The average PSNR improvement of different sub-sample accuracy techniques over the integer-sample FS using QCIF video sequences.

	NSP	ΔPSNR			
		Group A	Group B	Group C	Average
FS + FQSA	49	1.75	1.60	2.33	1.90
FS + HQSA	16	1.73	1.59	2.29	1.87
FS + HHSA	8	1.05	1.06	1.53	1.21
FS + CSM	0	0.69	0.88	1.66	1.08
FS + PSSA	0	0.82	0.96	1.73	1.17
FS + HPSSA	4	1.41	1.37	1.98	1.58

CHAPTER IV

A NOVEL TRUE-MOTION ESTIMATION ALGORITHM AND ITS APPLICATION TO MOTION-COMPENSATED TEMPORAL FRAME INTERPOLATION

4.1 Introduction

Motion estimation (ME) has a vital role in video coding and several video processing applications, such as denoising, deinterlacing, and frame rate up-conversion (FRC) or frame interpolation. It is employed to exploit the temporal correlation between video frames either to reduce the temporal redundancy for video coding applications or to improve the visual video quality for video processing applications.

One might argue that some of these video processing applications may potentially utilize the existing motion vectors (MVs) from the decoder via MV post-processing to keep the complexity low; however, this may not usually be a feasible option. This infeasibility could be due to either difficulty of using MVs or lack of available MVs. As video coding and video processing applications are often implemented separate intellectual properties (IPs) in hardware, it may be very difficult to share the MVs between decoder and other video processing applications due to bandwidth, latency, storage, and design specification reasons. Besides, some of these video processing applications may be employed either before the encoding or after the decoding, and some of them may be employed at both places; if it is employed before the encoding then MVs are not available, as a result ME needs to be performed. For example, FRC is employed only at the display side after the decoder; deinterlacing and de-noising, however, can be utilized in both places.

In addition, the achievable visual quality using MV post-processing may not be as

good as the quality that can be attained using true MVs since the received MVs are generally generated at the encoder by employing ME, which utilizes a block matching algorithm (BMA), to minimize the prediction errors instead of to track the projected object motion. Better image and video quality is obtained when the MVs that track the projected object motion are used in video processing applications, such as denoising, deinterlacing, and FRC. For example, to increase the frame rate MV post-processing may be preferred for resource-limited handheld devices, whereas ME algorithm can be employed for high visual video quality demanding applications, such as broadcasting, television and multimedia players [43, 44]. Besides, human visual system (HVS) tolerates artifacts better in small displays compared with large displays due to the increased angular resolution; hence, handheld devices can tolerate lower complexity inferior methods compared with large displays.

To distinguish ME schemes used in video processing applications from regular ME techniques used in video coding, they are usually termed as true-motion estimation (TME) algorithms to emphasize the fact that their objective is to track the projected object motion rather than to reduce the temporal redundancy[1]. It is not easy to accurately estimate motion since ME is an ill-posed problem [90,100]. Projected object motion usually results in a coherent motion vector field (MVF) except on the motion boundaries; however, regular ME algorithms may not necessarily produce a coherent MVF since their objective is to minimize the prediction error or number of bits required to code the prediction error. Hence, regular ME algorithms are susceptible to give wrong motion trajectories despite the fact that the resulting prediction error is very small. To obtain a coherent MVF, TME algorithms further impose smoothness constraints on MVs by using their spatiotemporally neighboring blocks besides employing regular ME.

[1]It is worth noting that although the MVs obtained from TME are called true-motion vectors (TMVs), some researches refer to the MVs obtained from the full search as TMV in the context of comparison with fast ME algorithms.

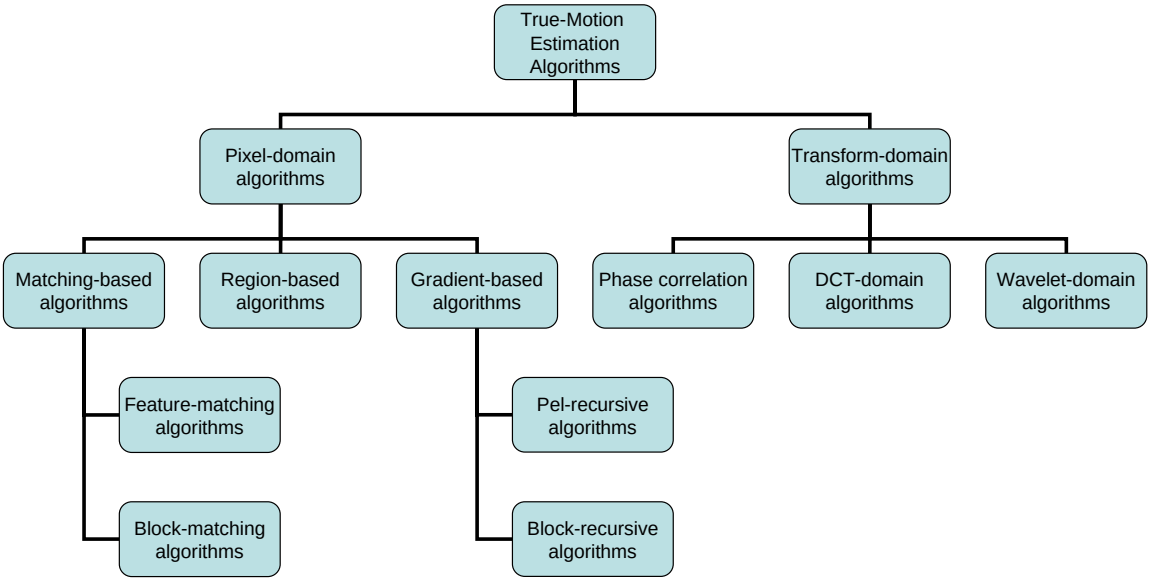


Figure 23: A classification of true-motion estimation algorithms.

TME algorithms can be broadly classified into pixel-domain and transform-domain algorithms as shown in Figure 23. Pixel-domain algorithms can be further classified into matching-based, gradient-based, and region-based algorithms. Matching-based algorithms include block matching and feature matching algorithms; gradient-based algorithms include pel-recursive and block-recursive algorithms; region-based algorithms include methods that segment frames into regions or objects having similar characteristics, such as motion, color, or texture. Transform-domain algorithms include phase correlation algorithms, DCT-domain matching algorithms, and wavelet-domain matching algorithms. Many researchers employed TME in various video applications, such as deinterlacing, denoising, FRC, and video coding [19, 20, 26, 27, 31–33, 46, 55, 57, 77, 82, 83, 89, 98], where BMA is employed for more efficient and hardware-friendly implementation. Some researchers employed other TME schemes as well, such as gradient-based, object-based, and transform-domain algorithms [13, 15, 24, 25, 28, 37, 62, 65, 68, 75, 93, 95, 97, 105].

Among the TME methods, BMA is the most widespread algorithm used in video coding and video processing applications partly due to its straightforward, simpler,

more efficient, and hardware-friendly implementation. BMAs target to minimize an objective function, which is also called block distortion measure (BDM), in the form

$$E_D(\boldsymbol{d}) = \sum_{\boldsymbol{n}\in\mathcal{B}} \rho\left(I_{k_1}[\boldsymbol{n}+\boldsymbol{d}], I_{k_2}[\boldsymbol{n}]\right), \tag{14}$$

where I_{k_1} and I_{k_2} are images at times k_1 and k_2, $\boldsymbol{d}$ is the displacement of $\boldsymbol{n}$ or MV at $\boldsymbol{n}$, $\mathcal{B}$ is the set of all pixels in I_{k_2}, and $\rho(\cdot,\cdot)$ is the matching criteria. Usually, $\rho(x,y) = |x-y|^p$ is used by setting $p=1$ for implementation simplicity, which is usually referred to as sum of absolute differences or errors (SAD or SAE). To achieve smoother MVFs, several different methods have been attempted in the literature. Imposing smoothness constraint can be done either explicitly or implicitly, or both. Approaches impose implicit smoothness constraint through the use of multi-resolution or predictive search for BMA. Enforcing explicit smoothness constraint is generally achieved by one of the following three ways: 1) adding a penalty term to (14), 2) using a different matching criteria, $\rho(\cdot,\cdot)$, and 3) post-processing the MVF obtained from applying (14).

Obtaining a robust TME algorithm for real-time video processing applications is a very challenging task. The algorithm should not only produce a smooth MVF but also satisfy some additional constraints for hardware implementation, such as algorithm regularity, low computational-complexity, and low memory bandwidth. Most of the reduction of the computational-complexity is usually achieved by decreasing the number of SAD calculations of the BMA through various fast ME strategies, such as reduction of search point, predictive search, and multi-resolution search. Similarly, memory bandwidth is affected by the block size, search range used in the algorithm, and the block processsing scan pattern [12].

In this paper, we propose a new TME algorithm that considers not only the computational complexity and regularity as in the aforementioned methods but also memory bandwidth; in addition, instead of relying on only a few of the spatiotemporal MV candidates for MVF smoothness, we exploit more spatiotemporal neighbors

by using a novel adaptive clustering algorithm to keep the number of predictors at a reasonable amount so that the number of SAD calculations remain feasible. Considering more neighbors increases the robustness of the algorithm as blocks may undergo quite different motions from neighboring blocks and some of the blocks may have inaccurate MVs. After MVF is obtained using TME, it can be used in the subsequent intended application. The accuracy of the MVF is more demanding in FRC compared to other video processing or coding applications. Hence, we further explore the use of the generated MVF for FRC. To obtain improved subjective and objective image quality compared with existing methods, that is also free of blocking artifacts we propose a novel method to obtain dense motion field at the interpolation instant and a motion-compensated temporal frame interpolation (MCTFI) method to generate the interpolated image.

The paper is organized as follows. In Section 4.2, we present a review of previous research on TME and MCTFI. The proposed TME algorithm is introduced in Section 4.3, and MCTFI is introduced in Section 4.4. Experimental results and discussion of the objective and subjective tests are presented in Section 4.5. Finally, Section 4.6 concludes the paper.

4.2 Previous Work

Performing TME has been the focus of video processing applications in the recent two decades, especially for FRC. Several algorithms have been proposed for TME. They obtain MVs that are closer to the projected object motion by imposing either explicit or implicit smoothness constraints, or a combination of both. Unlike explicit smoothness constraints, implicit smoothness constraints also help with speeding up the ME search. For instance, they are extensively employed to obtain fast ME algorithms in video coding [36].

An early successful TME implementation that targets smooth MVF is 3D Recursive Search (3DRS), in which spatiotemporally neighoring block MVs are used as MV candidates to accelerate the convergence of the algorithm and improve the MV correlation spatially among neighboring blocks [32]. This approach imposes smoothness constraint implicitly through predictive search and explicitly by using median filtering and penalty terms. Another method employing spatiotemporal MV candidates is Temporal Compensated ME with Simple Block-based Prediction (TC-SBP); it aims to further simplify 3DRS for hardware (HW) implementation by using less number of block matching calculation per block. TC-SBP uses three spatial and three temporal MV candidates for MV prediction, where temporal candidates are used to help speed up the convergence of the algorithm for the global motion field [98]; predictive search implicitly imposes smoothness constraint in this method. Although both of these TME algorithms guarantee smoother MVFs, they have difficulty in convergence for objects with quite different motions due to the limited reachable range. For this reason, it is recommended to perform couple of passes to help convergence; for example, three passes are recommended for 3DRS [11]. In addition, they have difficulty in convergence after scene changes due to the limited temporal candidates and reachable range. A recent method called Multi-pass and Motion Vector Propagation (MPMVP) improves the quality of both of these methods by employing multi-pass ME strategy along with variable block-sizes [89], where twelve passes are utilized. Performing multi-pass ME with variable block sizes is comparable to hierarchical ME approach; instead of obtaining lower resolutions, performing ME for fixed block sizes, and passing the coarser MV from lower resolution to higher resolution, this approach starts with larger block sizes initially (32×32) and reduces it incrementally to smaller (4×4) block sizes. MPMVP imposes smoothness constraint implicitly through multi-resolution search and explicitly by modifying the matching criteria. Although

employing multiple passes improves the accuracy of the MV considerably, it also increases the memory bandwidth manyfold, which is at a premium in multimedia SoCs as many IPs share the same bandwidth [12].

In another work, Ha et al. [39] aims to achieve more truthful motion trajectory by using overlapped block-based ME (OBME), where a larger region is used in (14) when evaluating the matching function for non-overlapped blocks, which corresponds to using implicit smoothness constraint through the use of modified matching criteria. To decrease the computational cost caused by larger size in matching, (14) is evaluated only at sub-sampled locations. In addition, a penalty term is used for uniform and periodical textures regions to bias the MV toward zero. Choi et al. [25] imposes smoothness by utilizing a penalty term and post-processing the MVF by median filter. In their later work [26], they only use a penalty term to impose smoothness. In a recent study, Wang et al. [96] uses the matching error of high-pass filtered images as additional penalty term to enforce the edge information.

Once the TME is performed, MVs can be used in MCTFI to obtain the interpolated frames. Use of MCTFI is based on the assumption that the object motion is translational and approximately linear over the duration between temporally adjacent frames. Usually, the decision on how to apply MC mandates the way ME is performed. For FRC applications, ME is usually employed in one of the following ways: 1) forward, 2) backward, 3) bidirectional; or a combination of them. For given two frames at times k_1 and k_2, these ME schemes result in MVF at k_1, k_2, or k_α ($k_1 < k_\alpha < k_2$), respectively. In bidirectional ME (BME), to be interpolated frame is partitioned into non-overlapping blocks, and the following minimization is performed to find a MV for each block

$$\boldsymbol{d}_\alpha = \arg\min_{\boldsymbol{d}} \sum_{\boldsymbol{n}\in\mathcal{B}} \rho\left(I_{k_1}[\boldsymbol{n}+\alpha\boldsymbol{d}], I_{k_2}[\boldsymbol{n}-(1-\alpha)\boldsymbol{d}]\right), \tag{15}$$

where $\boldsymbol{d}$ is limited to a search area, and $0 < \alpha < 1$.

If unidirectional ME (UME) is used, then the MVF at $k = k_\alpha$ is obtained either

by temporally shifting or projecting the MVF at k_1 or k_2 or both to k_α. Although temporally shifting the MVF guarantees that all blocks are assigned a MV at interpolation instant, it results in inaccurate MVs especially for large MV values. Projecting the MVF to k_α may result in areas with multiple MV assignment and no MV assignment, which are usually referred as overlap and hole regions, respectively. Previous attempts to handle overlap or hole regions consist of median filtering [25, 63, 92], object segmentation [45, 63, 92], spatial interpolation [54], hole region processing [50], and motion segmentation along-with mesh-based compensation [25]. Nonetheless, the required computational-complexity of these approaches is not low and as a result not very suitable for hardware implementation; besides, they could introduce undesired visual artifacts. In a different approach, Ojo et al. [76] propose a pixel-based nonlinear filtering to handle these regions implicitly. Although this method produces overall good quality interpolated pictures, it also produces halo artifacts in occlusion regions; Bellers et al. [9] proposed an improvement to reduce the halo artifacts by utilizing three consecutive frames in ME.

BME attracted attention in recent years mainly due to its invulnerability to overlap and hole regions [18,26,27,52,53,73,84,103]. BME and UME give consistent MVs for regions with simple translational motion; however, they have difficulty obtaining accurate MVs for regions with occlusion, rotation, or zoom. To obtain improved quality interpolated frames, different approaches employing a combination of BME and UME, forward and/or backward, are proposed [51, 71, 72, 88]. One important impediment of BME for FRC is that it has to be performed for each interpolated frame between two frames, which can be undesired if more than one interpolation frame is needed as in 24Hz to 60Hz conversion.

4.3 Proposed TME Algorithm

The proposed TME algorithm uses two consecutive frames and the previous MVF to estimate the current MVF. Similar to other practical video applications, it employs BMA to obtain a straightforward, efficient, and hardware-friendly implementation. To alleviate the computational-complexity of the full search (FS), BMAs usually employ fast ME (FME) algorithms. FME algorithms generally utilize one or a combination of the following categories: reduction of search points (SPs), simplification of matching criterion, multi-resolution search, predictive search, and fast FS [36]. Out of these categories, predictive search and multi-resolution search inherently impose smoothness constraints on MVF. Since multi-resolution search requires additional storage compared with predictive search we prefer predictive-search in our proposed scheme. There are plethora of algorithms for FME, we choose ENS algorithm due to its performance and ability to provide sub-sample accuracy at low computational-complexity [36]. ENS algorithm along with predictive search provides an effective FME that inherently impose smoothness constraint on MVF. In the following subsections, how explicit smoothness constraint is enforced, selection of the predictor MVs to impose implicit smoothness constraint, and the proposed true-motion estimation making use of both explicit and implicit smoothness constraints are explained.

4.3.1 Imposing Smoothness

Three different approaches were previously discussed in Section 4.1 for explicitly imposing smoothness, of which only the first two are suitable during the ME process. Using a modified matching function is usually motivated by intuition, whereas adding a penalty term can be theoretically justified by using a Bayesian MAP estimator [60, 100] which imposes certain prior distribution on the model parameters. The *a posteriori* probability distribution of the motion field

$$P(\mathcal{D}_k = \boldsymbol{d}_k | \mathcal{I}_k = I_k; I_{k-1}) \tag{16}$$

is used to obtain the MAP estimate by rewriting its Bayes equivalent as

$$P(\mathcal{I}_k = I_k|\mathcal{D}_k = \boldsymbol{d}_k; I_{k-1}) \cdot P(\mathcal{D}_k = \boldsymbol{d}_k; I_{k-1}), \tag{17}$$

where $\mathcal{D}_k$ is a vector random field, $\boldsymbol{d}_k$ is one of its realization, $\mathcal{I}_k$ is a scalar random field, and I_k is one of its realization. Then, the MAP estimate of $\boldsymbol{d}_k$ is computed as follows:

$$\widehat{\boldsymbol{d}}_k = \arg\max_{\boldsymbol{d}} \Big(P(\mathcal{I}_k = I_k|\mathcal{D}_k = \boldsymbol{d}_k; I_{k-1}) \cdot P(\mathcal{D}_k = \boldsymbol{d}_k; I_{k-1}) \Big), \tag{18}$$

where the first term is related to the observation model measuring how well $\boldsymbol{d}_k$ models the change, and the second term serves as a motion model explaining the prior information contribution of the random field $\mathcal{D}_k$, such as its smoothness.

To solve (18), it is assumed that the displaced frame difference (DFD) is a zero-mean Gaussian distribution; hence, the first term can be written as a product of zero-mean Gaussians. In addition, it is assumed that $\mathcal{D}_k$ is a MRF; so, the second term is a Gibbs distribution specified by cliques and a potential function. Using these assumptions and two-element cliques for MRF, (18) can be recast as

$$\widehat{\boldsymbol{d}}_k = \arg\min_{\boldsymbol{d}} \Big(\sum_{\boldsymbol{n}} |I_k[\boldsymbol{n}] - I_{k-1}[\boldsymbol{n} + \boldsymbol{d}]|^2 + \lambda \sum_{\boldsymbol{l} \in \mathcal{N}_{\boldsymbol{n}}} \|\boldsymbol{d}[\boldsymbol{n}] - \boldsymbol{d}[\boldsymbol{l}]\|^2 \Big) \tag{19}$$

where $\mathcal{N}_{\boldsymbol{n}}$ is a set of neighbors of $\boldsymbol{n}$ (e.g, 4- or 8-neighbors), λ is a constant, and $\|\cdot\|$ denotes Euclidian norm. For minimization of this function different approaches are proposed, such as simulated annealing, iterated conditional modes, highest confidence first [60]; however, they are very complex, and incompatible for real-time applications. In addition, even though this minimization gives rise to smooth MVF, it is inconvenient since the MVF is smooth even at the object boundaries.

A sub-optimal but real-time implementation can be obtained by use of BMA. Similar to other practical BMA algorithms, SAD can be used in the first term instead of sum of squared differences (SSD), which corresponds to assuming that DFD is a Laplacian distribution instead of Gaussian distribution. To prevent smoothing at the

object boundaries, edge information can be utilized in the second term. Potential function used in obtaining (19) is of the form

$$V(\boldsymbol{d}[\boldsymbol{n}], \boldsymbol{d}[\boldsymbol{l}]) = \|\boldsymbol{d}[\boldsymbol{n}] - \boldsymbol{d}[\boldsymbol{l}]\|^2, \quad \forall\, \boldsymbol{l} \in \mathcal{N}_{\boldsymbol{n}}, \tag{20}$$

which clearly treats each of the neighbors equally, and as a result smooths MVF even at the object boundaries.

In optical flow regularization, discontinuity preservation is achieved by anisotropic diffusion by limiting the smoothing at the object boundaries in the gradient direction proportional to the gradient value. Similarly, in BMA to preserve the discontinuity at object boundaries potential function can be modified accordingly. The potential function can be modified as

$$V(\boldsymbol{d}[\boldsymbol{n}], \boldsymbol{d}[\boldsymbol{l}]) = w(\boldsymbol{n}, \boldsymbol{l})\|\boldsymbol{d}[\boldsymbol{n}] - \boldsymbol{d}[\boldsymbol{l}]\|^2, \quad \forall\, \boldsymbol{l} \in \mathcal{N}_{\boldsymbol{n}}, \tag{21}$$

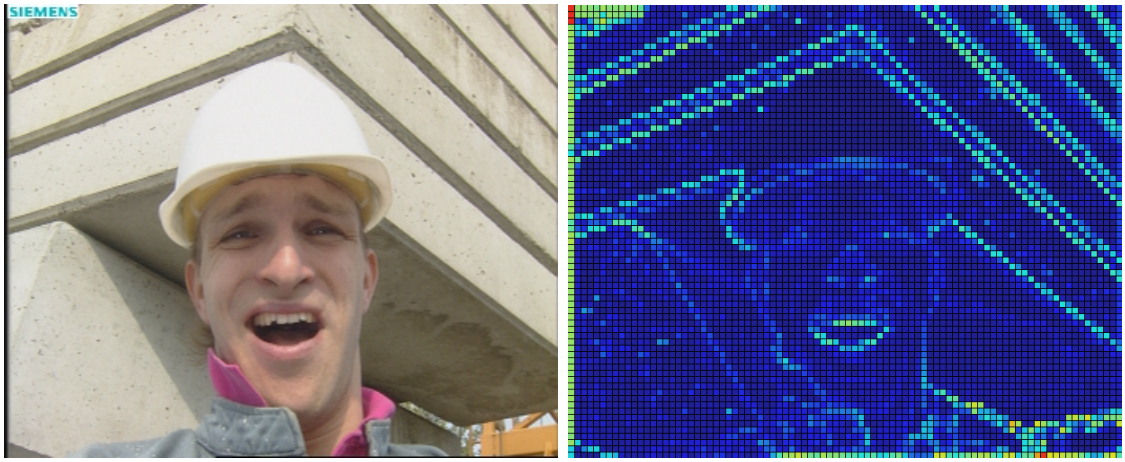
so that similar blocks in terms of edge content interact with each other more than other blocks. $w(\boldsymbol{n}, \boldsymbol{l})$ denotes a weight used to control the interaction between neighboring blocks. Maximum local variance (MLV) of a block can be used to measure the edge strength of a block [74]. Then, MLV for block $\boldsymbol{n}$ is defined as

$$g(\boldsymbol{n}) = \max_{i \in \boldsymbol{n}} \tfrac{1}{|\mathcal{N}_{\boldsymbol{i}}|+1} \sum_{j \in \mathcal{N}_{\boldsymbol{i}}} (j - \mu_i)^2 \tag{22}$$

where μ_i is the mean value of set $\{i \bigcup \mathcal{N}_i\}$, and $|\mathcal{N}_i|$ denotes the cardinality of $\mathcal{N}_i$. $g(\boldsymbol{n})$ is a nonnegative value denoting the edge strength of block $\boldsymbol{n}$. Figure 24 shows a sample frame from *Foreman* sequence and its corresponding edge strength map. Then, interaction between neighboring blocks can be defined similar to a Gaussian distribution as

$$w(\boldsymbol{n}, \boldsymbol{l}) = e^{-\frac{(g(\boldsymbol{n}) - g(\boldsymbol{l}))^2}{2\kappa_1^2}} \tag{23}$$

where κ_1 is the spread parameter used to control the amount of interaction between neighboring blocks. The range of $w(\boldsymbol{n}, \boldsymbol{l})$ is between [0,1]. If block $\boldsymbol{l}$ is similar to $\boldsymbol{n}$ in



(a) (b)

Figure 24: A sample edge strength map for a frame from *Foreman* sequence. (a) sample frame, (b) corresponding edge strength map for 4×4 block size, where edge strength values from low to high are mapped to colors from dark blue to light blue to green to yellow and finally to red.

terms of edge content, then $w(\boldsymbol{n}, \boldsymbol{l})$ will be close to one, otherwise to zero. This will prevent the continuity of MVF at object boundaries; instead, it will enable blocks to interact with similar blocks more and have more accurate and discontinuous MVs at object boundaries.

Additionally, reliability of the neighboring blocks' MV has to be taken into consideration when imposing smoothness explicitly. Even if a neighboring block is similar in terms of edge content, its MV may not be very accurate. To prevent imposing smoothness due to an inaccurate MV, corresponding smoothing term could be modified according to the accuracy of the MV. Neighboring block's BDM value can be used as a measure of its accuracy; if the BDM is lower than a threshold value then it is reliable, and its reliability decreases as the BDM value increases. For this purpose, weight $w(\boldsymbol{n}, \boldsymbol{l})$ is modulated with the reliability of the MV. The modified weight term becomes as follows:

$$w(\boldsymbol{n}, \boldsymbol{l}) = e^{-\frac{(g(\boldsymbol{n})-g(\boldsymbol{l}))^2}{2\kappa_1^2}} \min\left(1, e^{-\frac{E_D(\boldsymbol{l})-\mathrm{Th}}{2\kappa_2^2}}\right) \tag{24}$$

where κ_2 is the spread parameter used to control the amount of interaction between

neighboring blocks due to reliability. If the BDM value is less than or equal to the threshold value then the weight is not modified; once the value passes the threshold value the modification increases proportionally. Finally, a sub-optimal MAP estimate can be found by the following equation:

$$\widehat{\boldsymbol{d}}_k = \arg\min_{\boldsymbol{d}} \Big(\sum_{\boldsymbol{n}} |I_k[\boldsymbol{n}] - I_{k-1}[\boldsymbol{n}+\boldsymbol{d}]| + \lambda \sum_{\boldsymbol{l}\in\mathcal{N}_{\boldsymbol{n}}} w(\boldsymbol{n},\boldsymbol{l}) \|\boldsymbol{d}[\boldsymbol{n}] - \boldsymbol{d}[\boldsymbol{l}]\| \Big) \tag{25}$$

4.3.2 Predictor Selection

Predictive search employs MVs of spatiotemporally neighboring blocks to speed up the ME; equally, it increases the spatial correlation of MVs. Therefore, use of predictors implicitly imposes smoothness constraint on MVF. 3DRS uses two spatial and one temporal block as predictor. Based on the assumption that two different motions exist at motion edges, TC-SBP chooses three blocks that form a triangle such that current block is contained. MPMVP follows the same assumption and utilizes a multi-pass ME strategy, and in each pass uses a different set of predictors forming a triangle.

It's clear that for regions with single motion only one predictor suffices; and, for regions with complicated motion using three blocks may not capture the desired predictor. Utilizing MVs of all of the spatiotemporal neighbors as predictors will definitely improve the quality, although, at the cost of increased number of calculation. Besides, some of this calculation will be redundant since the MVF is consistent. To decrease the number of predictors to a manageable size, a clustering algorithm such as k-means can be used to consider only a subset of the MV predictors of the spatiotemporal neighbors.

K-means clustering is very fast, converges very rapidly, and its convergence is guaranteed [81]. So, it can be used to reduce the considered larger data points to much smaller number of candidates. However, it's not suitable for our case for two reasons: 1) it requires number of clusters to be fixed at the beginning, 2) number of iterations is not deterministic. For our case, MVs of the neighboring blocks may form

Algorithm 1: Adaptive Clustering Algorithm

Input: Sample set: $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$,
Number of maximum clusters: K,
Maximum cluster distance: D.
Output: Cluster means: $\mathbf{m}$,
Cluster labels: $\mathbf{c}$.

Initialize $\mathbf{m} = [\mathbf{0}, \emptyset, \cdots, \emptyset], \mathbf{c} = [1, \emptyset, \cdots, \emptyset]$;
Initialize $k = 1$; `/* # of formed clusters */`
foreach $\mathbf{x}_i \in \{\mathbf{x}_i\}_{i=2}^{N}$ **do**
    $\ell = \arg\min_{j \leq k} \|\mathbf{x}_i - \mathbf{m}_j\|$;
    $d_{\max} = \max_{j<i, \mathbf{c}_j = \ell} \|\mathbf{x}_i - \mathbf{x}_j\|$;
    if $d_{\max} \leq D$ **then**
        $\mathbf{c}_i = \ell$;
    else
        $k = \min(k + 1, K)$;
        $\mathbf{c}_i = k$;
    end
    Update $\mathbf{m}_{\mathbf{c}_i}$;
end

one or more clusters depending on the scene. For example, for static regions MVs will be grouped around zero resulting in one cluster. Depending on the region, we might have MVs in one, two, three, or four different directions; also, we might have outlier MVs. In addition, hardware implementation is not suitable for iterative processes if the number of iterations is not deterministic.

To cluster the MVs in the spatiotemporal neighborhood, it is desired to have a clustering algorithm that would adaptively give different number of clusters based on the maximum cluster distance specified at input, along with the additional constraint that this will take a fixed number of iterations. Hence, we modify the the k-means clustering algorithm to perform one iteration and form the clusters adaptively up to K clusters. Pseudocode of the proposed clustering algorithm is shown in Algorithm 1.

The algorithm takes the sample set $\mathcal{X}$, maximum number of clusters K, and the maximum cluster distance D as input; after execution it outputs the cluster means

m and cluster labels **c**. Since the sample set $\mathcal{X}$ is formed by zero MV and MVs of the spatiotemporal neighbors, zero MV is used in the initialization. Mean of the first cluster is set to zero and others to undefined. Similarly, label of the first sample is set to zero and others to undefined. Initally, number of formed clusters is set to one; depending on the displacement of MVs, the number will adaptively increase until K. Elements in the set, starting from second till last, are tested for the closest cluster and then the maximum distance to the elements of this cluster are calculated; if they are in the D proximity of a cluster, then they become a member of this cluster, otherwise they start a new cluster. Finally, cluster mean of the assigned cluster is updated and continued with the next sample in the set.

The output of this clustering algorithm serves as the predictor MVs for ENS algorithm. The number of maximum clusters K is set to six to account for near-stationary MV, possible four different directions, and outliers. Figure 25 shows a sample clustering result with six clusters. The maximum cluster distance D is set to 6 in accordance with the subsequent ENS algorithm. In addition, for distance calculation L^∞ norm is chosen since its unit ball for $\mathbb{R}^2$ is a square, which is more appropriate for the subsequent ENS algorithm as it can effectively find local minima in small square regions.

4.3.3 Estimating True-motion

In estimating the true-motion, both explicit and implicit smoothness constraints are employed in the proposed method. In performing TME using BMA for hardware implementation, not only computational complexity and regularity of the algorithm but also memory bandwidth is important. In several previous TME attempts, memory bandwidth has not been given higher priority in the design of the algorithm [32, 89]. For example, 3DRS requires three passes [11] and MPMVP requires twelve passes [89]

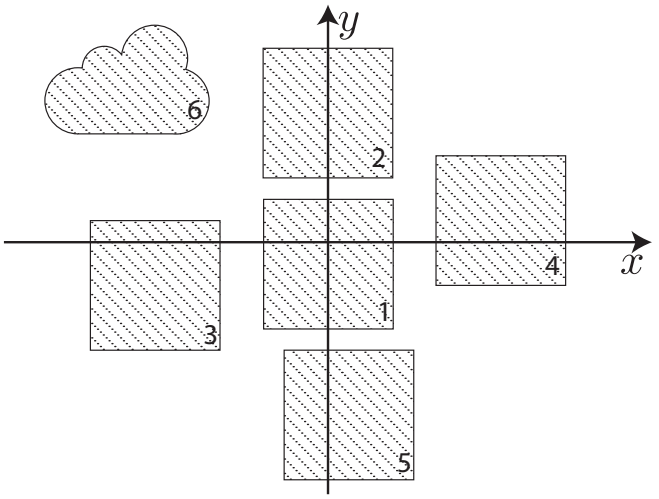


Figure 25: A sample clustering result.

for TME, which implies three- or twelve-fold increase of memory bandwidth, respectively.

In video processing applications the memory bandwidth requirement from the off-chip memory presents a serious impediment. It is even a bigger issue for multimedia SoCs as the off-chip memory is shared by a number of IPs [12]. To handle this requirement, Tuan et al. [94] propose four different levels of data reuse for ME depending on the reuse range: level-A, -B, -C, and -D. Level-C is the most practical one; however, it still requires manyfold memory bandwidth depending on the search range used in ME. To further handle this requirement, Beric et al. [12] employs a two-level memory hierarchy, which relies on the pixel data reuse properties of the underlying video processing algorithms. Sliding-L1 approach is the most practical both in terms of L1-cache area and off-chip memory bandwidth followed by stripe-L1 approach. Moreover, additional memory bandwidth saving can be obtained by using meandering scan instead of raster scan [12], in which consecutive lines are processed in reverse directions for increased reuse of pixel-data unlike raster scan where they are processed in the same direction; meandering scan also helps improve the implicit smoothness due to the use of reverse direction on alternating rows. In FRC, ME

and MCTFI drives the need for increased memory bandwidth. To keep the memory bandwidth at manageable levels not only ME should evade multiple pass algorithms but also pipelining should be employed to maximize the reuse of pixel-data for ME and MCTFI modules.

The proposed TME algorithm employs explicit and implicit smoothness constraints in using BMA, namely, ENS algorithm [36]. For FRC, ENS algorithm is employed for ME in two stages; for 8×8 and 4×4 blocks, in order. In addition, meandering scan is used for reduced memory bandwidth and improved implicit smoothness. The following subsections give the details for each stage.

4.3.3.1 ME for 8×8 *blocks*

For each 8×8 block, predictor MVs are obtained using the proposed adaptive clustering algorithm as outlined in Algorithm 1. In this clustering algorithm, MVs of the spatiotemporally neighboring blocks shown in Figure 26 and zero MV are used as input. In addition to the colocated block and its 8-neighbors in the previous frame, four additional diagonal blocks are used to improve the temporal convergence [32]. In the current frame, only the causal blocks of the 8-neighbors and block at $(j-1, i+2)$ are used; for right-to-left scan direction these positions are vertically mirrored. The resulting clustering output will have at least one or at most K predictor MVs. In addition, a predictor MV for global motion is used for faster convergence of large global motions. A low complexity global ME method is used to obtain this predictor; it divides the current and reference frames into $N_1 \times M_1$ grid of sub-images and calculates their horizontal and vertical projections to find a global MV [64] for each direction independently. Then, the predictor MV giving the minimum cost is selected as the winner and ME is performed around this point for additional search points.

For ease of explanation, the set of k-pixels away 8-neighbors of $\boldsymbol{n}$ is defined as

$$\mathcal{N}^k_{\boldsymbol{n},8} = \{(x,y)|x = -k, 0, k\ ;\ y = -k, 0, k\} \setminus \{(0,0)\}, \tag{26}$$

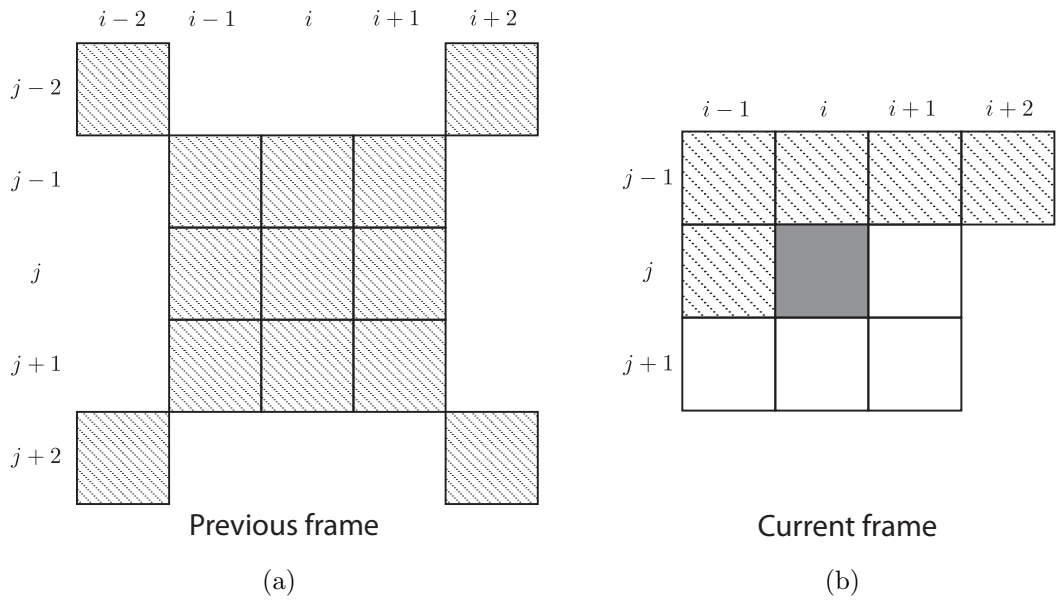
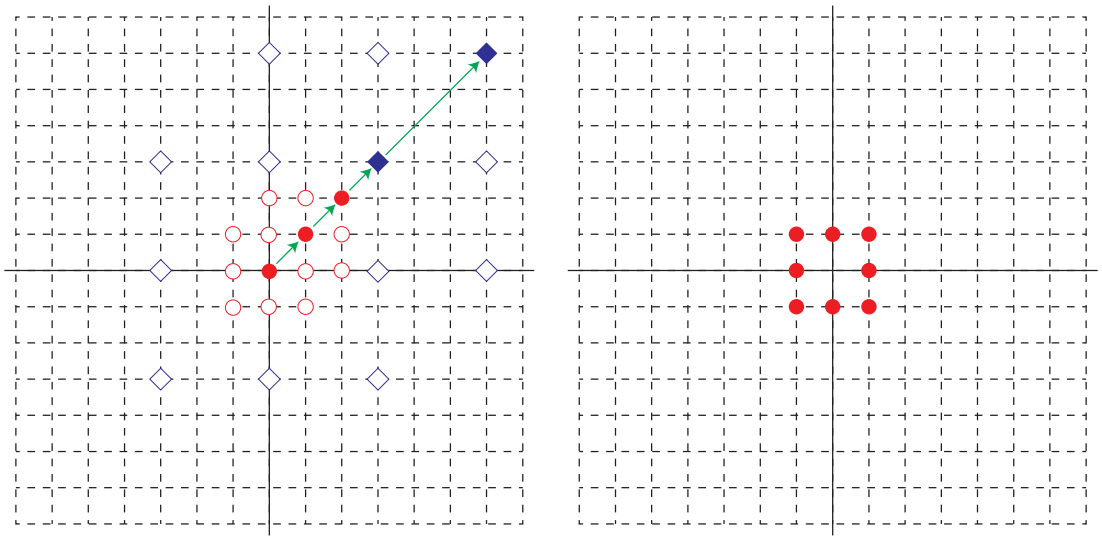


Figure 26: Predictor set for 8×8 blocks. (a) Predictors on previous frame. (b) Predictor on current frame.

where (x, y) denotes the rectangular coordinates relative to the location of a pixel of interest $\boldsymbol{n}$. First, points in $\mathcal{N}^1_{\boldsymbol{n},8}$ of the search center are checked, and the minimum BDM point is found. Then, the remaining $\mathcal{N}^1_{\boldsymbol{n},8}$ of the minimum BDM point are checked; depending on the location of the minimum cost being at the center of a side or at a corner of the previous $k \times k$ grid, either additional three or five points are checked, respectively. Second, points in $\mathcal{N}^3_{\boldsymbol{n},8}$ of the search center are checked, and the minimum BDM point is found. Then, the remaining $\mathcal{N}^3_{\boldsymbol{n},8}$ of this minimum BDM point are checked. A possible set of check points is shown in Figure 27(a), where empty or filled circles and diamonds indicate the check points or minimum BDM points, respectively, at each step. The resulting search point with the minimum cost is the coarse MV of this block. Subsequently, it is further refined for 4×4 blocks as explained in the following subsection.



(a) (b)

Figure 27: Search points of ENS algorithm. (a) A possible set of search points of ENS algorithm for 8×8 blocks. (b) Set of search points of ENS algorithm for 4×4 blocks.

4.3.3.2 ME for 4×4 blocks

For each 4×4 block, predictor MVs are obtained from the coarse MVs of 8×8 blocks obtained in the previous stage. Each 4×4 block considers its 8-neighbors. Since some of the neighbors fall into the same 8×8 block, only a subset of them could be considered. Diagonal neighbors uniquely represent this subset. Figure 28 shows the diagonal neighbors of a 4×4 block at each possible quadrant location of a 8×8 block. Considering only the diagonal neighbors of each 4×4 block gives four predictor MVs. Out of these four predictor MVs, the one giving the minimum cost is chosen as the winner, and ME is performed on neighboring eight locations as shown in Figure 27(b). Using the cost of eight neighboring locations, MV with sub-sample accuracy (SSA) can be obtained without using the costly sub-sample ME [36].

At the end of these two stages, MV with SSA is obtained for each 4×4 block.

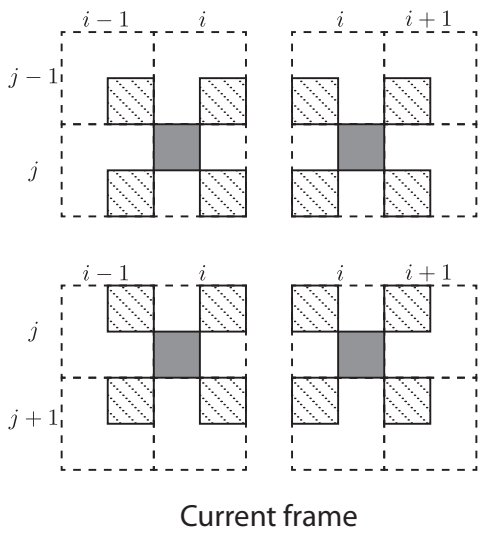


Figure 28: Predictor set for 4×4 blocks for each possible quadrant location. Gray block denotes the current 4×4 block, and patterned blocks denote its predictor blocks.

4.4 Motion-compensated Frame Interpolation

Applying TME, as explained in Section 4.3, gives MVs with SSA for each 4×4 block. As discussed in Section 4.2, ME can be employed in one of the following ways: 1) forward, 2) backward, 3) bidirectional; or a combination of them. To enable better quality multiple interpolation frames between two existing frames UME is employed instead of BME; to better handle occlusion regions effectively for better quality both forward and backward TME are used. Occluded areas exist in one of the two successive frames; depending on these areas being covering or uncovering corresponding former or latter frame has the necessary information, and the interpolated frame should be constructed accordingly. In addition, to prevent blocking artifacts in the interpolated frame, dense motion field at the interpolation instant will be obtained and used for temporal interpolation. In the following subsections, obtaining dense motion field and performing interpolation using both forward and backward dense motion fields are explained.

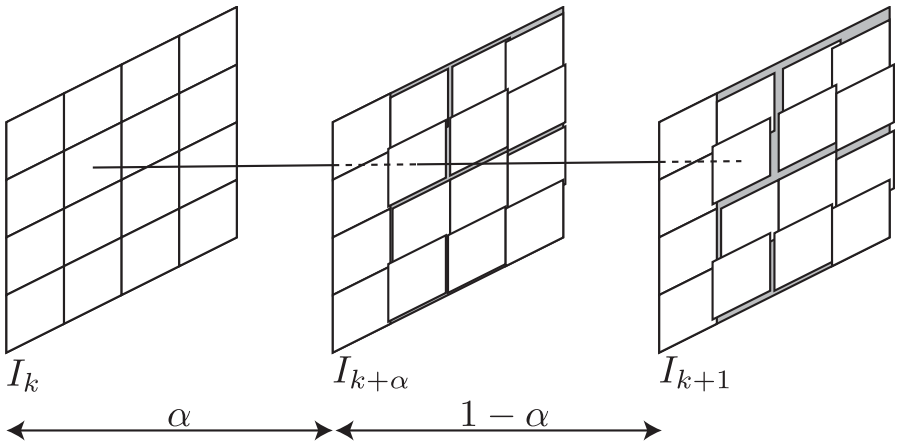


Figure 29: Unidirectional ME and its projection to the intermediate frame.

4.4.1 Obtaining Dense Motion Field

Applying backward and forward TME between frames I_k and I_{k+1} will result in MVFs $\boldsymbol{d}^b_{k+1}$ and $\boldsymbol{d}^f_k$, respectively, where superscript denotes the direction. Although temporally shifting them to obtain the MVFs at $k+\alpha$ is more straightforward compared with projecting them, it gives inaccurate MVs especially for large values. Hence, these MVFs will be projected to $k+\alpha$ for better interpolated image quality. Although the use of UME results in a contiguous MVF at its associated anchor frame, the projected MVF at interpolation instant may not be contiguous. In fact, it will have overlap and hole regions due to differing MV values of neighboring blocks as illustrated in Figure 29.

Projecting the MVFs $\boldsymbol{d}^b_{k+1}$ and $\boldsymbol{d}^f_k$ to $\boldsymbol{d}^b_{k+\alpha}$ and $\boldsymbol{d}^f_{k+\alpha}$, respectively, will result in overlaps due to areas with multiple MV assignment. Out of multiple MVs passing through the same block, choosing the one with lower BDM gives better interpolated image quality. When the projection of multiple MVs intersect at a block of interpolated frame at $k+\alpha$, the size of overlap may range from 1 to $|\mathcal{B}_{i,j}|$ pixels. To minimize the number of overlaps, block $\mathcal{B}_{i,j}$ can be partitioned into small blocks by assigning the same MV before projection. A 4×4 block can be partitioned into 4 or 16 equal size blocks. Although partitioning into pixel resolution results in the smallest area

overlaps, partitioning into 2×2 blocks is a better trade-off between quality and complexity considering the subsequent filtering step. Each forward and backward MV for a 2×2 block is first assigned its parent's MV and then projected as follows:

$$\boldsymbol{d}^b_{k+\alpha}\big[\boldsymbol{b_{i,j}} + (1-\alpha)\boldsymbol{d}^b_{k+1}[\boldsymbol{b_{i,j}}]\big] = \boldsymbol{d}^b_{k+1}[\boldsymbol{b_{i,j}}], \tag{27}$$

$$\boldsymbol{d}^f_{k+\alpha}\big[\boldsymbol{b_{i,j}} + \alpha\boldsymbol{d}^f_{k}[\boldsymbol{b_{i,j}}]\big] = \boldsymbol{d}^f_{k}[\boldsymbol{b_{i,j}}], \tag{28}$$

where $\boldsymbol{b_{i,j}}$ denotes the index of each 2×2 block.

After all the blocks are projected, some of the 2×2 blocks of the MVFs $\boldsymbol{d}^b_{k+\alpha}$ and $\boldsymbol{d}^f_{k+\alpha}$ may not have any MV assignment, and the resulting holes of the MVFs may be at different sizes. 2-dimensional (2-D) filtering is used to assign MVs to these blocks as

$$\boldsymbol{d}_{k+\alpha}[\boldsymbol{b_{i,j}}] = \frac{\sum_{\boldsymbol{b_{l,m}}\in\mathcal{N}} \Psi[\boldsymbol{b_{l,m}}]\boldsymbol{d}_{k+\alpha}[\boldsymbol{b_{l,m}}]}{\sum_{\boldsymbol{b_{l,m}}\in\mathcal{N}} \Psi[\boldsymbol{b_{l,m}}]}, \tag{29}$$

where superscript is omitted since the equation is the same for both forward and backward MVFs, and Ψ is used to denote the reliability of each block. Ψ is defined as

$$\Psi[\boldsymbol{b_{l,m}}] = \begin{cases} 0 & \text{if } \boldsymbol{d}_{k+\alpha}[\boldsymbol{b_{l,m}}] \text{ is a hole,} \\ E_D(\boldsymbol{b_{l,m}}) & \text{if } \boldsymbol{d}_{k+\alpha}[\boldsymbol{b_{l,m}}] \text{ is not a hole,} \end{cases} \tag{30}$$

where $E_D(\boldsymbol{b_{l,m}})$ is the BDM of block $\boldsymbol{b_{l,m}}$. For lower complexity implementation all MVs in a neighborhood can be treated equally by using $E_D(\boldsymbol{b_{l,m}}) = 1$ instead of BDM. In the proposed method a 5×5 window is used for 2-D filtering along with uniform weights. It is worth noting that in some of the sequences it is possible that a hole region larger than filter support may exist; and, this results in the denominator of (29) to be zero. Although it is possible to implement a preemptive approach in software, such as using a larger support for the filter or eroding the MVF to close the remaining holes, these may not be feasible for a hardware approach. Alternatively, the MV will be assigned to zero if the denominator of (29) equals zero for a hardware friendly implementation.

After MV projection and hole filling, all of the 2×2 blocks of the MVFs $\boldsymbol{d}^b_{k+\alpha}$ and $\boldsymbol{d}^f_{k+\alpha}$ will have a MV assignment. To further refine the MVs to pixel level to obtain the dense motion field, bilinear interpolation is employed. Resulting dense motion field is used to obtain the interpolated frame.

4.4.2 Interpolation

Obtained dense motion fields at the interpolation instant are used to generate the interpolated frame. Existence of occlusion areas between neighboring frames complicates the interpolation process since they may exist in either one of them. Occlusion areas are either covering or uncovering depending on the former or latter frame has the necessary information. An uncovering area at frame $k+1$ does not have its correspondence at frame k, hence the backward MVs in this area may not be accurate. To accurately find the motion of this occlusion area frame $k+2$ has to be used as it would have the corresponding area. Similarly, for an covering area at frame k, frame $k-1$ has to be utilized to find the corresponding MV of this covering area. Hence, to effectively handle occlusion regions between two frames, one additional frame in each direction is required, resulting in total four frames. This, however, increases the complexity and storage requirements of the solution. To enable a low-complexity solution, we employ only two neighboring frames and gracefully obtain the interpolated images.

To handle the occlusion areas using two frames, some methods employ occlusion detection mechanisms and use their detection results to interpolate the occlusion area by selecting the pixels from the corresponding neighboring frame depending on the area is covering or uncovering. However, such hard switching methods at pixel or block level may produce strong local temporal artifacts that are very annoying during the playback. It's usually preferred to have a global degeneration than a strong local distortion [32]. As a result, occlusion areas are dealt with implicitly using both

forward and backward MVFs. In the proposed method, hard switching is avoided by employing soft decision for occlusion areas. Both forward and backward MVFs are used to generate the interpolated frame by mixing the forward and backward interpolated frames inversely proportional to their motion-compensated error as

$$\begin{aligned} I_{k+\alpha}[\boldsymbol{x}] &= \frac{\dfrac{I^f_{k+\alpha}[\boldsymbol{x}]}{1+\Delta^f[\boldsymbol{x}]} + \dfrac{I^b_{k+\alpha}[\boldsymbol{x}]}{1+\Delta^b[\boldsymbol{x}]}}{\dfrac{1}{1+\Delta^f[\boldsymbol{x}]} + \dfrac{1}{1+\Delta^b[\boldsymbol{x}]}} \\ &= \frac{\big(1+\Delta^b[\boldsymbol{x}]\big) I^f_{k+\alpha}[\boldsymbol{x}] + \big(1+\Delta^f[\boldsymbol{x}]\big) I^b_{k+\alpha}[\boldsymbol{x}]}{2+\Delta^b[\boldsymbol{x}]+\Delta^f[\boldsymbol{x}]}, \end{aligned} \tag{31}$$

where $I^b_{k+\alpha}[\boldsymbol{x}]$ and $I^f_{k+\alpha}[\boldsymbol{x}]$ denote backward and forward interpolated pixel values, and $\Delta^b[\boldsymbol{x}]$ and $\Delta^f[\boldsymbol{x}]$ denote their corresponding motion-compensated error value that is defined as

$$\Delta^b[\boldsymbol{x}] = \sum_{\boldsymbol{l}\in\mathcal{N}_{\boldsymbol{x}}} \big| I_k\big[\boldsymbol{l} + \alpha \boldsymbol{d}^b_{k+1}[\boldsymbol{l}]\big] - I_{k+1}\big[\boldsymbol{l} - (1-\alpha)\boldsymbol{d}^b_{k+1}[\boldsymbol{l}]\big] \big|, \tag{32}$$

$$\Delta^f[\boldsymbol{x}] = \sum_{\boldsymbol{l}\in\mathcal{N}_{\boldsymbol{x}}} \big| I_k\big[\boldsymbol{l} - \alpha \boldsymbol{d}^f_{k}[\boldsymbol{l}]\big] - I_{k+1}\big[\boldsymbol{l} + (1-\alpha)\boldsymbol{d}^f_{k}[\boldsymbol{l}]\big] \big|, \tag{33}$$

where for neighborhood $\mathcal{N}_{\boldsymbol{x}}$ a 3×3 window is used instead of pixel difference to increase the robustness when calculating the motion-compensated error value for each pixel. In (31), one is added to the motion-compensated error to prevent possible division by zero error.

Backward and forward interpolated pixel values are obtained from the neighboring frames using both forward and backward MVFs as follows:

$$I^f_{k+\alpha}[\boldsymbol{x}] = (1-\alpha) I_k\big[\boldsymbol{x} - \alpha \boldsymbol{d}^f_k[\boldsymbol{x}]\big] + \alpha I_{k+1}\big[\boldsymbol{x} + (1-\alpha)\boldsymbol{d}^f_k[\boldsymbol{x}]\big], \tag{34}$$

$$I^b_{k+\alpha}[\boldsymbol{x}] = (1-\alpha) I_k\big[\boldsymbol{x} + \alpha \boldsymbol{d}^b_{k+1}[\boldsymbol{x}]\big] + \alpha I_{k+1}\big[\boldsymbol{x} - (1-\alpha)\boldsymbol{d}^b_{k+1}[\boldsymbol{x}]\big], \tag{35}$$

4.5 Results and Discussion

Assessment of FRC is not an easy task. Although it is desirable to have an objective assessment approach to compare frame interpolation techniques for video, unfortunately there is not any accepted objective criterion in the literature that gives meaningful results. Instead of measuring the quality of the overall video, current approach adopted by researchers is to measure the quality of the interpolated frames as follows: first, temporally down-sample the video sequence, usually by two; second, interpolate these down-sampled frames using FRC techniques; third, compare with the original video frames using objective metrics, such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) [101].

The distortion introduced into image/video can be classified into three categories in regard to the human visual perception as sub-, near-, and supra-threshold distortions [70, 104]. Sub-threshold distortion is below just-noticeable difference (JND) and usually imperceivable by the HVS; near-threshold distortion is slightly above JND; and, supra-threshold distortion generally appears in a structured form, such as blockiness, ghosting, and blurring. While PSNR has good performance on measuring near-threshold distortion, SSIM has good performance on measuring both near- and supra-threshold artifacts [104]. Since supra-threshold dominates the human visual perception, SSIM is a more meaningful measure than PSNR in terms of the HVS. However, PSNR is more commonly used for comparison in the literature. PSNR simply measures the mean squared error; however, SSIM is constructed by incorporating luminance, contrast, and structure information at each pixel locality. Therefore, both PSNR and SSIM values are used for comparison with recent state-of-the-art methods proposed in the literature. In addition, the overall video quality is assessed subjectively for the temporal artifacts. The proposed algorithm has been successfully tested on a variety of video sequences; only a few of the results are shown in this paper.

Eleven video sequences of common intermediate format (CIF) size (352×288) and

one video sequence of HD720p size (1280×720) are used for in this paper, which are widely available in the video processing literature. These sequences are: *News, Stefan, Foreman, Mother & Daughter, Mobile, Highway, Football (b), Tt, Garden, Paris, Container, and Crew.* The proposed algorithm is compared with recently proposed state-of-the art FRC algorithms presented by Kang et al. [51] and Wang et al. [96] et al., and several other existing algorithms they used for comparison [26, 30, 66, 69]. In addition, a professional video production suite, Apple Final Cut Studio (Apple FCS), is used for comparison [6]; specifically, *Motion 4* application in Apple FCS is used to perform MCTFI. Apple FCS uses optical flow for MCTFI to generate high-quality interpolated frames and does not generate blocking artifacts as other methods used for comparison [26, 30, 51, 66, 69, 96]. Also, it should be noted that since Apple FCS works with video in RGB format, test sequences have been converted between YCbCr and RGB formats so that MCTFI can be applied by Apple FCS. To prevent the mismatch due to color conversion in PSNR and SSIM calculation, the original sequence went through the same color format conversion between YCbCr and RGB as well.

4.5.1 Objective Assessment

Comparison of the objective measures PSNR and SSIM of the proposed algorithm against the method presented by Kang et al. [51] and other algorithms they used for comparison [26,66,69] and Aplle FCS [6] are shown in Table 4. Number of frames used in comparison is shown in third column of the table along with the resolution of each sequence. The method of Kang et al. achieves an average maximum improvement of PSNR and SSIM by 1.62 dB and 0.025, respectively, compared to other methods. Apple FCS and the proposed method substantially improve the values compared to the method of Kang et al. [51]; Apple FCS gives an average increase in PSNR and SSIM by 3.6 dB and 0.061, respectively; and, the proposed method gives an average

increase in PSNR and SSIM by 4.36 dB and 0.07, respectively.

Comparison of the objective measures PSNR and SSIM of the proposed algorithm against the method presented by Wang et al. [96] and other algorithms they used for comparison [26, 30] and Aplle FCS [6] are shown in Table 5. Number of frames used in comparison is shown in second column of the table for each sequence. The method of Wang et al. achieves an average maximum improvement of PSNR by 4.58 dB compared to other methods. Apple FCS and the proposed method substantially improve the values compared to the method of Wang et al. [96]; Apple FCS gives an average increase in PSNR by 1.68 dB; and, the proposed method gives an average increase in PSNR by 3.66 dB.

In summary, Apple FCS and the proposed method give significantly improved objective results compared to existing MCTFI algorithms. In addition, the proposed method gives slightly better results than Apple FCS. On average, the proposed method gives an average increase in PSNR and SSIM by 0.7 dB and 0.009, respectively, in sequences shown in Table 4; 2 dB and 0.026, respectively, in sequences shown in Table 5.

4.5.2 Subjective Assessment

Subjective assessment of up-converted video sequences supplements the quantitative assessment given above in Tables 4 and 5. Existing MCTFI methods have poor subjective video quality mainly due to blocking artifacts. Unlike the existing methods, both Apple FCS and the proposed algorithm do not suffer from the blocking artifacts mainly due to the use of dense motion field at the interpolation instant. Hence, the proposed method is compared against the Apple FCS output for subjective quality. When both videos are played, the proposed algorithm offers comparable or better quality than Apple FCS; snapshots of videos are shown in Figures 30 and 31, and are

available at the specified filename. To support this conclusion, some of the interpolated frames of Apple FCS and the proposed method, and their corresponding SSIM maps are given in Figures 32, 33, 34, 35, 36, 37, 38, and 39 for tested video sequences.

SSIM values range between 0 and 1 inclusive; 0 denotes bad and 1 denotes good match, which correspond to dark and bright pixel values, respectively. To show SSIM maps of Y, Cb, and Cr channels together in one image, SSIM values of both chroma channels are transformed by $1-\frac{x}{2}$; consequently, combined SSIM value of both chroma channels corresponds to the first quadrant of the Cb-Cr plane and each Cb and Cr channel SSIM values can be shown with shades of red and blue colors, respectively, and their combination. SSIM maps given in Figures 32, 33, 34, 35, 36, 37, 38, and 39 show combined SSIM maps of Y, Cb, and Cr for each interpolated frame.

MC-FRC results of some of the interpolated frames for different sequences obtained by Apple FCS and the proposed method are shown in Figures 32, 33, 34,35, 36, 37, 38, and 39. In both figures, columns correspond to different video sequences and rows correspond to (in order): original frame, interpolated frame of Apple FCS, interpolated frame of the proposed method, SSIM map of the Apple FCS interpolated frame, and SSIM map of the proposed method interpolated frame.

For frame 20 of *Tt* sequence, the proposed method improves PSNR by 3.4 dB and SSIM by 0.058 compared with Apple FCS. As can be seen from the SSIM maps, Apple FCS gives bad match around the arm and the ball, whereas the proposed method limits the bad match only to boundary.

For frame 78 of *Foreman* sequence, the proposed method improves PSNR by 0.84 dB and SSIM by 0.019 compared with Apple FCS. Apple FCS causes ghosting around nose, ear, hat, and collar. Overall, bad matches in the interpolated frame of the proposed method is much less than the interpolated frame of Apple FCS.

For frame 60 of *Stefan* sequence, the proposed method improves PSNR by 2.567 dB and SSIM by 0.042 compared with Apple FCS. Apple FCS causes text "harman" on

the sign to break up and boundary line to get deform. In addition, bad match around the player and left and right boundary of the image is more in Apple FCS interpolated frame than the interpolated frame of the proposed method.

For frame 58 of *Mobile* sequence, the proposed method improves PSNR by 0.803 dB and SSIM by 0.008 compared with Apple FCS. Apple FCS causes ghosting on last row of the numbers on calendar and slightly worse match around the ball.

For frame 50 of *Paris* sequence, the proposed method improves PSNR by 3.195 dB and SSIM by 0.022 compared with Apple FCS. Apple FCS causes bad match around man's head, green books behind his head, books behind woman's head, and woman's hands. Overall, bad matches in the interpolated frame of Apple FCS are more prominent than the interpolated frame of the proposed method.

For frame 18 of *Highway* sequence, the proposed method improves PSNR by 5.213 dB and SSIM by 0.023 compared with Apple FCS. Apple FCS causes some parts of left line marking to have double image and gives bad match on some parts of the barrier.

For frame 4 of *Football (b)* sequence, the proposed method improves PSNR by 1.423 dB and SSIM by 0.125 compared with Apple FCS. Unlike the proposed method, in addition to bad matches around the boundary of players Apple FCS causes bad matches on the grass around the players. This particularly creates very annoying temporal artifacts when the video is played.

For frame 172 of *Crew* sequence, the proposed method improves PSNR by 2.631 dB and SSIM by 0.022 compared with Apple FCS. Unlike the proposed method, in addition to bad matches around astronaut's hand Apple FCS causes bad matches on logo prints in the neighborhood of hand.

In summary, the proposed method gives better match than Apple FCS. This advantage mainly comes from the accuracy of the motion estimation. Since the proposed algorithm employs both implicit and explicit smoothness constraints in a

way to prevent the continuity of MVF at object boundaries it can successfully obtain TMVs in the frame without over-smoothing the MVF. Also, use of UME in forward and backward direction, obtaining dense motion field at the interpolation instant, and gracefully obtaining the interpolated frame using both forward and backward MVs help prevent the blocking artifacts and handle occlusions implicitly without creating halo.

4.5.3 Complexity Analysis

Both computational complexity and memory bandwidth are critical for real-time hardware and software implementations. Computational complexity comparison of MC-FRC methods can be assessed by comparing the number of SAD calculations performed in ME. The decision of using unidirectional or bidirectional ME algorithm impacts not only the memory bandwidth utilization but also computational complexity. To better understand the comparison of memory bandwidth utilization by different MC-FRC methods, consider an up-conversion ratio of r.

In terms of number SAD calculations, the proposed method has much lower computational complexity than the benchmark methods of Wang et al. [96] and Kang et al. [51]. The proposed method uses a FME in conjunction with predictive search not only to decrease the number of search points but also to enforce implicit smoothness on the MVF. However, both of the benchmark methods use the full search algorithm; as a result, they have much more number of calculations per block.

Similarly, the proposed method has a much lower memory bandwidth utilization than the benchmark methods. Since they make use of the bidirectional ME, ME has to be performed $r - 1$ times, once for each interpolated frame, for an up-conversion ratio of r; whereas, the proposed method performs forward and backward ME once for each pair of input frames regardless of the up-conversion ratio r.

4.6 Conclusion

A novel true-motion estimation algorithm and its application to MC-FRC is presented. Computational-complexity, regularity, and memory bandwidth is considered when designing the algorithm so that a low-complexity implementation in ASIC or FPGA can be easily achieved. The proposed TME algorithm imposes implicit and explicit smoothness constraints on BMA to obtain more accurate MVs with SSA representing the projected object motion. Obtained MVs from forward and backward ME are refined to lower block sizes and projected to the interpolation frame instant. Resulting MVs at the interpolation instant are efficiently handled for overlaps and holes to obtain a dense motion field. Finally, forward and backward dense motion fields are used to gracefully obtain the interpolated frame so that blocking artifacts are prevented and occlusions are implicitly handled without creating halo.

Objective and subjective assessment of the proposed method shows that it gives better results in comparison to other existing algorithms. Obtained interpolated frames are free from blocking artifacts, provide higher picture quality, and are visually more pleasing than others; in addition, perceptual temporal quality is also better than the existing methods when the resulting video sequence is played. Unlike some algorithms, it does not introduce flicker, which is a crucial requirement for video applications.

Figure 30: Subjective MC-FRC results of the proposed method and the Apple FCS for sequences: (a,b) *container*, (c,d) *crew720p*, (e,f) *football_b*, (g,h) *foreman*, (i,j) *garden*, and (k,l) *highway*. Suffixes *_pro* and *_fcs* in the filenames refer to the proposed method and the Apple FCS results, respectively.
(a) (dikbas_salih_201112_phd_container_pro.mov, 41M)
(b) (dikbas_salih_201112_phd_container_fcs.mov, 42M)
(c) (dikbas_salih_201112_phd_crew720p_pro.mov, 268M)
(d) (dikbas_salih_201112_phd_crew720p_fcs.mov, 265M)
(e) (dikbas_salih_201112_phd_football_b_pro.mov, 34M)
(f) (dikbas_salih_201112_phd_football_b_fcs.mov, 36M)
(g) (dikbas_salih_201112_phd_foreman_pro.mov, 40M)
(h) (dikbas_salih_201112_phd_foreman_fcs.mov, 42M)
(i) (dikbas_salih_201112_phd_garden_pro.mov, 56M)
(j) (dikbas_salih_201112_phd_garden_fcs.mov, 56M)
(k) (dikbas_salih_201112_phd_highway_pro.mov, 260M)
(l) (dikbas_salih_201112_phd_highway_fcs.mov, 242M)

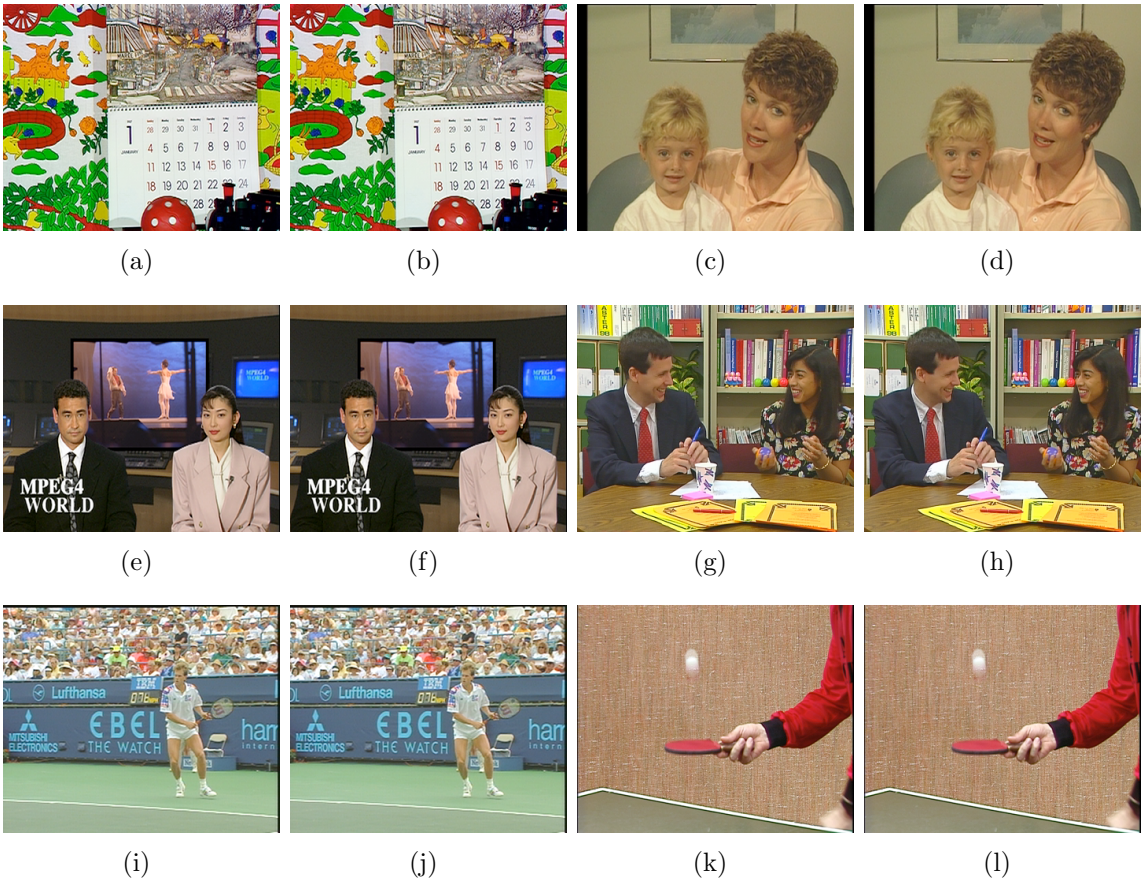
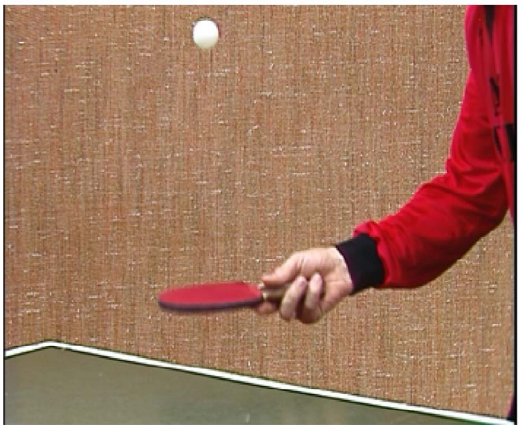


Figure 31: Subjective MC-FRC results of the proposed method and the Apple FCS for sequences: (a,b) *mobile*, (c,d) *mother*, (e,f) *news*, (g,h) *paris*, (i,j) *stefan*, and (k,l) *tt*. Suffixes *_pro* and *_fcs* in the filenames refer to the proposed method and the Apple FCS results, respectively.
(a) (dikbas_salih_201112_phd_mobile_pro.mov, 58M)
(b) (dikbas_salih_201112_phd_mobile_fcs.mov, 61M)
(c) (dikbas_salih_201112_phd_mother_pro.mov, 32M)
(d) (dikbas_salih_201112_phd_mother_fcs.mov, 34M)
(e) (dikbas_salih_201112_phd_news_pro.mov, 39M)
(f) (dikbas_salih_201112_phd_news_fcs.mov, 40M)
(g) (dikbas_salih_201112_phd_paris_pro.mov, 176M)
(h) (dikbas_salih_201112_phd_paris_fcs.mov, 182M)
(i) (dikbas_salih_201112_phd_stefan_pro.mov, 45M)
(j) (dikbas_salih_201112_phd_stefan_fcs.mov, 48M)
(k) (dikbas_salih_201112_phd_tt_pro.mov, 45M)
(l) (dikbas_salih_201112_phd_tt_fcs.mov, 46M)



(a)

(b) 32.501dB, 0.910

(c) 35.896dB, 0.968

(d)

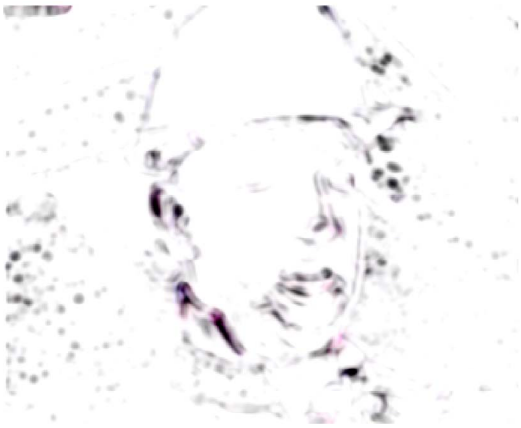
(e)

Figure 32: Objective MC-FRC results for frame 20 of *Tt* sequence: a) original, b) output of the Apple FCS, c) output of the proposed method, d) SSIM map of the Apple FCS output, e) SSIM map of the proposed method.

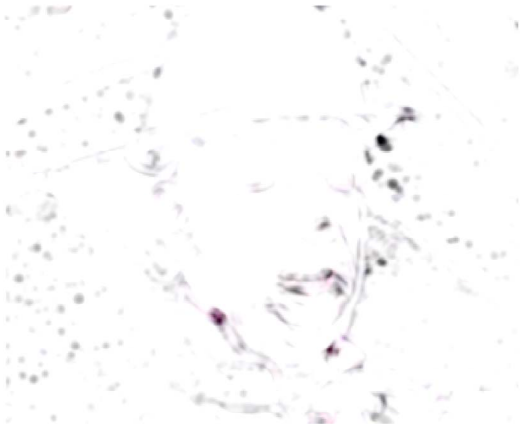
(a)

(b) 33.905dB, 0.931

(c) 34.745dB, 0.950



(d)



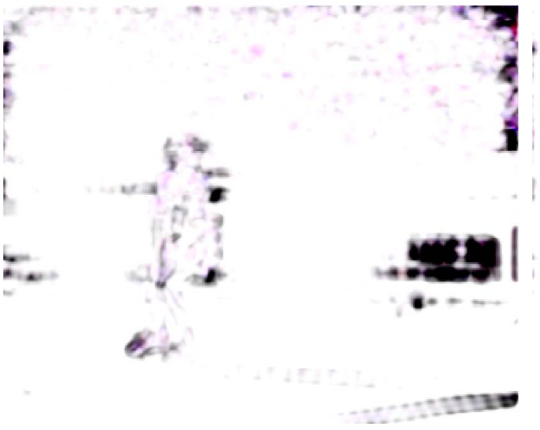
(e)

Figure 33: Objective MC-FRC results for frame 78 of *Foreman* sequence: a) original, b) output of the Apple FCS, c) output of the proposed method, d) SSIM map of the Apple FCS output, e) SSIM map of the proposed method.

(a)

(b) 27.905dB, 0.923

(c) 30.472dB, 0.965



(d)

(e)

Figure 34: Objective MC-FRC results for frame 60 of *Stefan* sequence: a) original, b) output of the Apple FCS, c) output of the proposed method, d) SSIM map of the Apple FCS output, e) SSIM map of the proposed method.

(a)

(b) 29.040dB, 0.957

(c) 29.843dB, 0.965

(d)

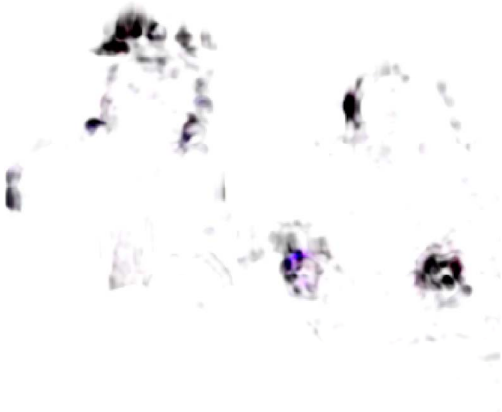
(e)

Figure 35: Objective MC-FRC results for frame 58 of *Mobile* sequence: a) original, b) output of the Apple FCS, c) output of the proposed method, d) SSIM map of the Apple FCS output, e) SSIM map of the proposed method.

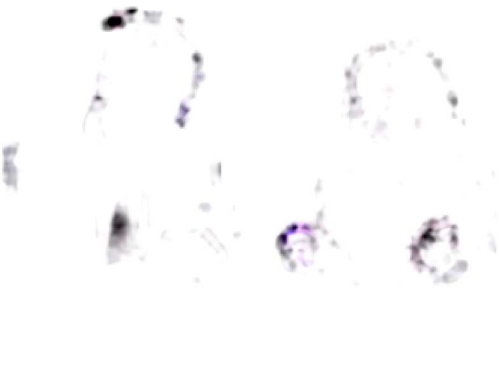
(a)

(b) 31.011dB, 0.953

(c) 34.206dB, 0.975



(d)



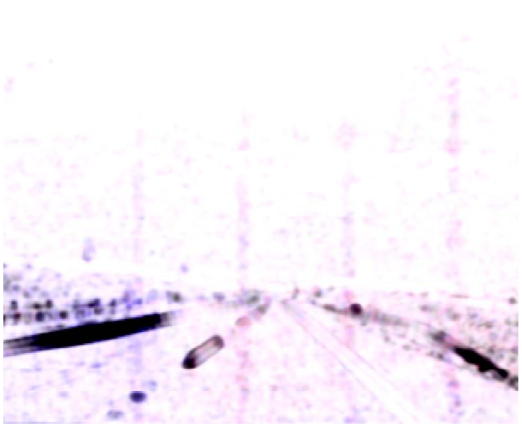
(e)

Figure 36: Objective MC-FRC results for frame 50 of *Paris* sequence: a) original, b) output of the Apple FCS, c) output of the proposed method, d) SSIM map of the Apple FCS output, e) SSIM map of the proposed method.

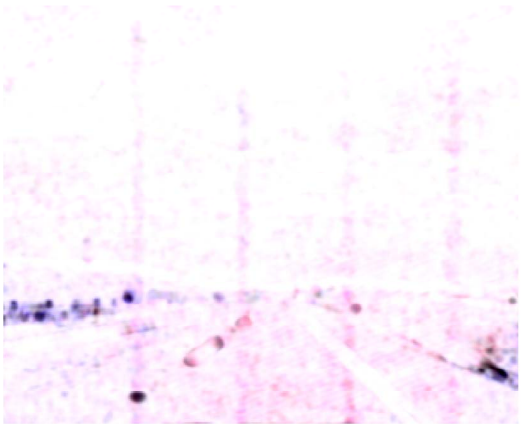
(a)

(b) 30.502dB, 0.909

(c) 35.715dB, 0.932



(d)



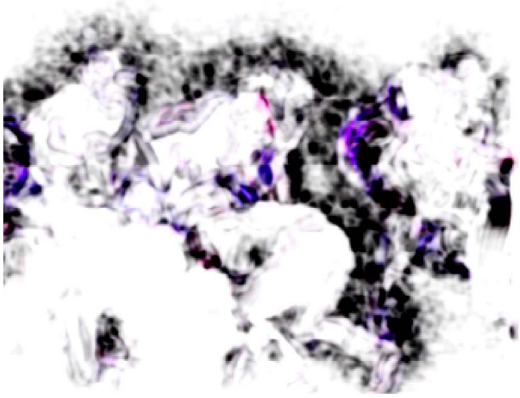
(e)

Figure 37: Objective MC-FRC results for frame 18 of *Highway* sequence: a) original, b) output of the Apple FCS, c) output of the proposed method, d) SSIM map of the Apple FCS output, e) SSIM map of the proposed method.

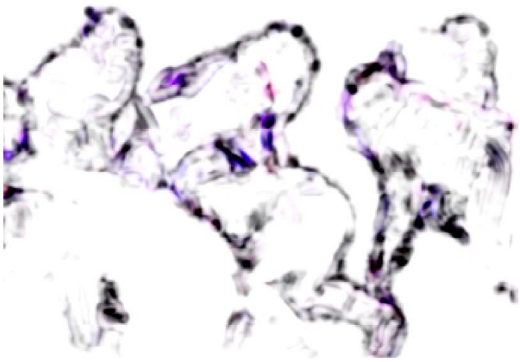
(a)

(b) 25.653dB, 0.763

(c) 27.076dB, 0.888



(d)



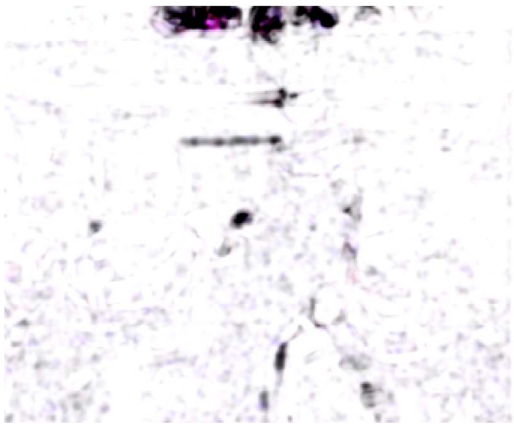
(e)

Figure 38: Objective MC-FRC results for frame 4 of *Football (b)* sequence: a) original, b) output of the Apple FCS, c) output of the proposed method, d) SSIM map of the Apple FCS output, e) SSIM map of the proposed method.

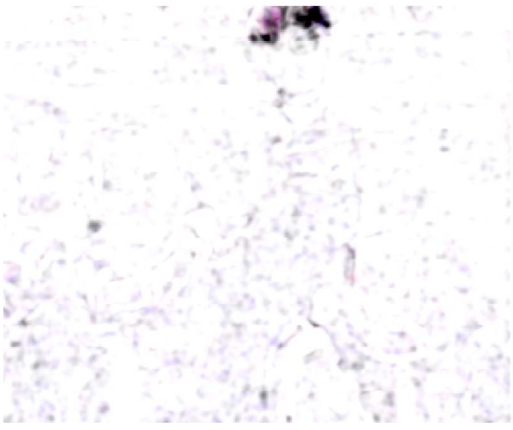
(a)

(b) 32.071dB, 0.903

(c) 34.702dB, 0.925



(d)



(e)

Figure 39: Objective MC-FRC results for cropped frame 172 of *Crew* sequence: a) original, b) output of the Apple FCS, c) output of the proposed method, d) SSIM map of the Apple FCS output, e) SSIM map of the proposed method.

Table 4: Objective measure comparison of the proposed algorithm with existing algorithms using CIF/720p video sequences. PSNR denotes the Peak Signal-to-Noise Ratio and SSIM denotes the structural similarity index.

			Ref. [26]		Ref. [69]		Ref. [66]		Ref. [51]		Apple FCS [6]		Proposed	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
News	CIF	(90)	32.884	0.965	32.969	0.966	34.671	0.974	34.946	0.975	36.873	0.979	**38.214**	**0.984**
Stefan	CIF	(90)	23.885	0.875	23.770	0.870	23.048	0.825	24.324	0.888	27.869	0.913	**29.157**	**0.942**
Foreman	CIF	(300)	27.243	0.832	28.171	0.846	28.557	0.846	29.191	0.865	32.584	0.914	**33.252**	**0.940**
Mother_daughter	CIF	(300)	37.405	0.949	36.607	0.944	38.105	0.953	38.726	0.958	42.245	0.976	**42.708**	**0.980**
Mobile	CIF	(300)	21.401	0.842	22.541	0.872	22.549	0.873	23.113	0.890	28.608	0.951	**28.994**	**0.957**
Highway	CIF	(300)	28.973	0.784	30.645	0.791	29.966	0.795	31.245	0.805	32.998	0.919	**33.450**	**0.922**
Crew	720p	(300)	26.734	0.771	27.115	0.764	26.937	0.768	28.339	0.816	34.141	**0.969**	**34.662**	0.962
Average			28.361	0.860	28.831	0.865	29.119	0.862	29.983	0.885	33.617	0.946	**34.348**	**0.955**

Table 5: Objective measure comparison of the proposed algorithm with existing algorithms using CIF video sequences. PSNR denotes the Peak Signal-to-Noise Ratio and SSIM denotes the structural similarity index.

		Ref. [30]		Ref. [26]		Ref. [96]		Apple FCS [6]		Proposed	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Football (b)	(260)	19.71	–	19.69	–	22.74	–	23.033	0.654	**24.140**	**0.746**
Tt	(300)	25.69	–	25.38	–	29.58	–	31.798	0.917	**34.920**	**0.956**
Garden	(374)	22.14	–	22.56	–	26.96	–	33.061	0.974	**34.368**	**0.976**
Mobile	(300)	21.59	–	19.57	–	25.09	–	28.608	0.951	**28.994**	**0.957**
Paris	(1065)	27.80	–	28.05	–	33.53	–	33.819	0.975	**36.341**	**0.984**
Container	(300)	35.32	–	38.65	–	41.87	–	39.518	0.980	**42.995**	**0.988**
Average		25.38	–	25.65	–	29.96	–	31.640	0.909	**33.626**	**0.935**

CHAPTER V

CONCLUSION AND FUTURE WORK

5.1 Contributions

In this dissertation, a low-complexity solution to motion-compensated video frame rate up-conversion is proposed. In obtaining this solution, the following have been achieved.

- We presented a novel low-complexity fast motion estimation (FME) algorithm to be used in frame rate up-conversion (FRC), which can be used in other video processing and coding applications as well. To reduce the computational complexity, we have proposed a FME algorithm capable of producing MVs with SSA. Unlike existing FME algorithms, the proposed FME algorithm considers both search point reduction and interpolation-free sub-sample ME simultaneously. The proposed FME algorithm is designed in such a way that the block distortion measure (BDM) is modeled as a parametric surface in the vicinity of the integer-sample motion vector to enable low computational complexity sub-sample motion estimation without pixel interpolation. Experimental results on video test sequences show that the proposed FME algorithm substantially reduces computational complexity integer- and sub-sample ME considerably compared with traditional methods at the cost of negligible MSE degradation from the FS.

- A novel true-motion estimation algorithm and its application to MC-FRC is presented. Computational-complexity, regularity, and memory bandwidth is considered when designing the algorithm so that a low-complexity implementation in ASIC or FPGA can be easily achieved. The proposed TME algorithm

imposes implicit and explicit smoothness constraints on BMA to obtain more accurate MVs with SSA representing the projected object motion. Obtained MVs from forward and backward ME are refined to lower block sizes and projected to the interpolation frame instant. Resulting MVs at the interpolation instant are efficiently handled for overlaps and holes to obtain a dense motion field. Finally, forward and backward dense motion fields are used to gracefully obtain the interpolated frame so that blocking artifacts are prevented and occlusions are implicitly handled without creating halo. Objective and subjective assessment of the proposed method shows that it gives better results in comparison to other existing algorithms. Obtained interpolated frames are free from blocking artifacts, provide higher picture quality, and are visually more pleasing than others. In addition, perceptual temporal quality is also better than the existing methods when the resulting video sequence is played. Unlike some algorithms, it does not introduce flicker, which is a crucial requirement for video applications.

5.2 Future Research Directions

Real-time applications generally employ FME algorithms similar to the one presented in this dissertation. These algorithms are usually heuristic search strategies that are supported with comparable performance results. To the best of our knowledge, no published work is reported that gives a unified framework for generating and evaluating such search strategies. A unified framework will enable pre-assessment between different search strategies before doing any simulation to compare number of search points or PSNR for different video sequences. Derivative-free optimization techniques [29] along with lattices and lattice sampling offers necessary tools for building such a framework.

The proposed TME algorithm employs explicit and implicit smoothness constraints in using BMA to give smoother MVF. The obtained MVF can be used in other video processing and coding applications, such as denoising, deinterlacing, and video compression, to get improved results as well. In real-time applications motion-adaptive temporal noise filtering (TNF) is usually employed; motion-compensated TNF can use the MVs from TME for better filtering performance. Similarly, deinterlacing algorithm can utilize motion-compensation to obtain improved performance with respect to motion-adaptive deinterlacing. State-of-the-art video compression standards utilize rate-distortion optimization to minimize the bit-rate at a given quality level, which is better than just minimizing the prediction error. As the proposed TME is an MAP estimate, and the MAP estimate of $\boldsymbol{d}$ is equivalent to minimum description length estimate [7], the proposed TME can be employed in video encoding as a lower complexity alternative to rate-distortion optimization.

The quality of the proposed MC-FRC can be further improved by using more frames than two. Use of more frames will enable better handling of the occlusion regions and object boundaries. Using both forward and backward TME results for three frames will significantly improve the robustness of the algorithm at the expense of increased implementation complexity.

REFERENCES

[1] "Information technology-coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s-part 2: Video," *JTC1/SC29/WG11, ISO/IEC 11 172-2 (MPEG-1 Video)*, 1993.

[2] "Video codec for audiovisual services at p×64 kbit/s," tech. rep., ITU-T SG15, ITU-T Rec. H.261, 2 ed., 1993.

[3] "Information technology-coding of audio visual objects-part 2 visual," tech. rep., JTC1/SC29/WG11, ISO/IEC 14 469-2 (MPEG-4 Visual), 2000.

[4] "Video coding for low bit rate communication," tech. rep., ITU-T SG16, ITU-T Rec. H.263, 3rd ed., 2000.

[5] "Draft ITU-T rec. and final draft international standard of joint video specification (ITU-T Rec. H.264-ISO/IEC 14 496-10 AVC)," tech. rep., Joint Video Team (JVT) of ITU-T and ISO/IEC JTC1, Geneva, JVT of ISO/IEC MPEG and ITU-T VCEG, JVT-G050r1, 2003.

[6] Apple Inc., "Final Cut Studio." http://www.apple.com/finalcutstudio/, Jan. 2011.

[7] Barron, A., Rissanen, J., and Yu, B., "The minimum description length principle in coding and modeling," *IEEE Trans. Inf. Theory*, vol. 44, pp. 2743–2760, Oct. 1998.

[8] Barten, P. G. J., "Formula for the contrast sensitivity of the human eye," vol. 5294, pp. 231–238, SPIE, 2003.

[9] Bellers, E., van Gurp, J., Janssen, J., Braspenning, J., and Wittebrood, R., "Solving occlusion in frame-rate up-conversion," in *Consumer Electronics, 2007. ICCE 2007. Digest of Technical Papers. International Conference on*, pp. 1–2, Jan. 2007.

[10] Berbecel, G., *Digital Image Display: Algorithms and Implementation*. Wiley-SID series in display technology, Wiley, 2003.

[11] Beric, A., de Haan, G., van Meerbergen, J., and Sethuraman, R., "Towards an efficient high quality picture-rate up-converter," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 2, pp. 363–366, Sept. 2003.

[12] Beric, A., van Meerbergen, J., de Haan, G., and Sethuraman, R., "Memory-centric video processing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, pp. 439–452, Apr. 2008.

[13] Biswas, M. and Nguyen, T., "A novel motion estimation algorithm using phase plane correlation for frame rate conversion," in *Signals, Systems and Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on*, vol. 1, pp. 492–496, Nov. 2002.

[14] Blume, H., "Vector-based nonlinear upconversion applying center-weighted medians," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* (E. R. Dougherty, J. T. Astola, & H. G. Longbotham, ed.), vol. 2662 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pp. 142–153, Mar. 1996.

[15] Blume, H., Herczeg, G., Erdler, O., and Noll, T. G., "Object based refinement of motion vector fields applying probabilistic homogenization rules," *IEEE Trans. Consum. Electron.*, vol. 48, pp. 694–701, Aug. 2002.

[16] Burns, R. W., *Television: an international history of the formative years.* History of technology series, The Institution of Engineering and Technology, 1998.

[17] Castagno, R., Haavisto, P., and Ramponi, G., "A method for motion adaptive frame rate up-conversion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 436–446, Oct. 1996.

[18] Chen, T., "Adaptive temporal interpolation using bidirectional motion estimation and compensation," in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 2, pp. 313–316, 2002.

[19] Chen, Y.-K. and Kung, S., "Rate optimization by true motion estimation," in *Multimedia Signal Processing, 1997., IEEE First Workshop on*, pp. 187–194, June 1997.

[20] Chen, Y.-R. and Tai, S.-C., "True motion-compensated de-interlacing algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, pp. 1489–1498, Oct. 2009.

[21] Cheung, C.-H. and Po, L.-M., "A novel cross-diamond search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 1168–1177, Dec. 2002.

[22] Cheung, C.-H. and Po, L.-M., "Novel cross-diamond-hexagonal search algorithms for fast block motion estimation," *IEEE Trans. Multimedia*, vol. 7, pp. 16–22, Feb. 2005.

[23] Chiew, T.-K., Chung-How, J., Bull, D., and Canagarajah, C., "Interpolation-free subpixel refinement for block-based motion estimation," *Proc. SPIE - Int. Soc. Opt. Eng. (USA)*, vol. 5308, no. 1, pp. 1261–1269, 2004.

[24] Cho, Y.-H., Lee, H.-Y., Park, D.-S., and Kim, C.-Y., "Enhancement for temporal resolution of video based on multi-frame feature trajectory and occlusion compensation," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pp. 389–392, Nov. 2009.

[25] Choi, B.-D., Han, J.-W., Kim, C.-S., and Ko, S.-J., "Frame rate up-conversion using perspective transform," *IEEE Trans. Consum. Electron.*, vol. 52, pp. 975–982, Aug. 2006.

[26] Choi, B.-D., Han, J.-W., Kim, C.-S., and Ko, S.-J., "Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, pp. 407–416, Apr. 2007.

[27] Choi, B.-T., Lee, S.-H., and Ko, S.-J., "New frame rate up-conversion using bi-directional motion estimation," *IEEE Trans. Consum. Electron.*, vol. 46, pp. 603–609, Aug. 2000.

[28] Choi, B.-T., Lee, S.-H., Park, Y.-J., and Ko, S.-J., "Frame rate up-conversion using the wavelet transform," in *Consumer Electronics, 2000. ICCE. 2000 Digest of Technical Papers. International Conference on*, (Los Angles, CA), pp. 172–173, June 2000.

[29] Conn, A. R., Scheinberg, K., and Vicente, L. N., *Introduction to Derivative-Free Optimization*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2009.

[30] Dane, G. and Nguyen, T., "Optimal temporal interpolation filter for motion-compensated frame rate up conversion," *IEEE Trans. Image Process.*, vol. 15, pp. 978–991, Apr. 2006.

[31] de Haan, G., "IC for motion-compensated de-interlacing, noise reduction, and picture-rate conversion," *IEEE Trans. Consum. Electron.*, vol. 45, pp. 617–624, Aug. 1999.

[32] de Haan, G., Biezen, P. W. A. C., Huijgen, H., and Ojo, O. A., "True-motion estimation with 3-D recursive search block matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 368–379, Oct. 1993.

[33] de Haan, G. and Biezen, P., "An efficient true-motion estimator using candidate vectors from a parametric motion model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 85–91, Feb. 1998.

[34] de Haan, G., *Video Processing for Multimedia Systems*. Eindhoven, 2000.

[35] den Boer, W., *Active Matrix Liquid Crystal Displays*. Electronics & Electrical, Elsevier, 2005.

[36] Dikbas, S., Arici, T., and Altunbasak, Y., "Fast motion estimation with interpolation-free sub-sample accuracy," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, pp. 1047–1051, July 2010.

[37] Fu, M. F., Au, O., and Chan, W. C., "Temporal interpolation using wavelet domain motion estimation and motion compensation," in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 3, pp. 393–396, 2002.

[38] Giunta, G. and Mascia, U., "Estimation of global motion parameters by complex linear regression," *IEEE Trans. Image Process.*, vol. 8, pp. 1652–1657, Nov. 1999.

[39] Ha, T., Lee, S., and Kim, J., "Motion compensated frame interpolation by new block-based motion estimation algorithm," *IEEE Trans. Consum. Electron.*, vol. 50, pp. 752–759, May 2004.

[40] Hayashi, S., Asakura, R., Nakamura, J., Takahashi, M., and Karaki, N., *Nikkei Microdevices' Flat Panel Display: 2006 Yearbook*. InterLingua Educational Pub., 2006.

[41] Hill, P., Chiew, T., Bull, D., and Canagarajah, C., "Interpolation free subpixel accuracy motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, pp. 1519–1526, Dec. 2006.

[42] Hilman, K., Park, H. W., and Kim, Y., "Using motion-compensated frame-rate conversion for the correction of 3:2 pulldown artifacts in video sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 869–877, Sept. 2000.

[43] Huang, A.-M. and Nguyen, T., "A multistage motion vector processing method for motion-compensated frame interpolation," *IEEE Trans. Image Process.*, vol. 17, pp. 694–708, May 2008.

[44] Huang, A.-M. and Nguyen, T., "Correlation-based motion vector processing with adaptive interpolation scheme for motion-compensated frame interpolation," *IEEE Trans. Image Process.*, vol. 18, pp. 740–752, Apr. 2009.

[45] Huang, C.-L. and Chao, T.-T., "Motion-compensated interpolation for scan rate up-conversion," *Optical Engineering*, vol. 35, no. 1, pp. 166–176, 1996.

[46] Huska, J. and Kulla, P., "A new recursive search with multi stage approach for fast block based true motion estimation," in *Radioelektronika, 2007. 17th International Conference*, pp. 1–6, Apr. 2007.

[47] Huynh-Thu, Q. and Ghanbari, M., "Temporal aspect of perceived quality in mobile video broadcasting," *IEEE Trans. Broadcast.*, vol. 54, pp. 641 –651, Sept. 2008.

[48] Ibrahim, K. F., *Newnes Guide to Television and Video Technology*. Elsevier Science & Technology, 2007.

[49] Jain, J. R. and Jain, A. K., "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. 29, pp. 1799–1808, Dec. 1981.

[50] Jeon, B.-W., Lee, G.-I., Lee, S.-H., and Park, R.-H., "Coarse-to-fine frame interpolation for frame rate up-conversion using pyramid structure," *IEEE Trans. Consum. Electron.*, vol. 49, pp. 499–508, Aug. 2003.

[51] Kang, S.-J., Yoo, S., and Kim, Y. H., "Dual motion estimation for frame rate up-conversion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, pp. 1909–1914, Dec. 2010.

[52] Kang, S.-J., Cho, K.-R., and Kim, Y. H., "Motion compensated frame rate up-conversion using extended bilateral motion estimation," *IEEE Trans. Consum. Electron.*, vol. 53, pp. 1759–1767, Nov. 2007.

[53] Kang, S.-J., Yoo, D.-G., Lee, S.-K., and Kim, Y., "Multiframe-based bilateral motion estimation with emphasis on stationary caption processing for frame rate up-conversion," *IEEE Trans. Consum. Electron.*, vol. 54, pp. 1830–1838, Nov. 2008.

[54] Kaup, A. and Aach, T., "Efficient prediction of uncovered background in interframe coding using spatial extrapolation," in *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, vol. 5, pp. 501–504, Apr. 1994.

[55] Kawaguchi, K. and Mitra, S. K., "Frame rate up-conversion considering multiple motion," in *Image Processing, 1997. Proceedings., International Conference on*, vol. 1, (Santa Barbara, CA), pp. 727–730, Oct. 1997.

[56] Kelly, D. H., "Visual responses to time-dependent stimuli. i. amplitude sensitivity measurements," *J. Opt. Soc. Am.*, vol. 51, pp. 422–429, Apr. 1961.

[57] Kim, J.-S. and Kim, L.-S., "Noise robust motion refinement for motion compensated noise reduction," *IEICE Transactions on Information and Systems*, vol. E91-D, pp. 1581–1583, May 2008.

[58] Koga, T., Iinuma, K., Hirano, A., Iijima, Y., and Ishiguro, T., "Motion-compensated interframe coding for video conferencing," in *Proc. Nat. Telecommun. Conf.*, (New Orleans, LA), pp. G5.3.1–5.3.5, Nov. 29–Dec. 3 1981.

[59] Kolb, H., Nelson, R., Fernandez, E., and Jones, B. W., "Webvision: The organization of the retina and visual system." http://webvision.med.utah.edu, Apr. 2011.

[60] Konrad, J., "Motion detection and estimation," in *Handbook of Image and Video Processing* (Bovik, A., ed.), chapter 3.10, pp. 253–274, Burlington, MA, USA: Elsevier Academic Press, 2nd ed., 2005.

[61] Krauskopf, J., "Effect of retinal image stabilization on the appearance of heterochromatic targets," *J. Opt. Soc. Am.*, vol. 53, pp. 741–741, June 1963.

[62] Krishnamurthy, R., Woods, J. W., and Moulin, P., "Frame interpolation and bidirectional prediction of video using compactly encoded optical-flow fields and label fields," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 713–726, Aug. 1999.

[63] Kuo, T.-Y. and Kuo, C.-C., "Motion-compensated interpolation for low-bit-rate video quality enhancement," vol. 3460, (USA), pp. 277–288, 1998.

[64] Lee, W. P., Belt, H. J. W., and van der Tol, E., "Video frame rate conversion for mobile devices," vol. 5684, pp. 32–42, SPIE, 2005.

[65] Lee, W. H., Choi, Y., Choi, K., and Ra, J. B., "Frame rate up conversion via image fusion based on variational approach," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pp. 885–888, Sept. 2010.

[66] Lee, Y.-L. and Nguyen, T., "Method and architecture design for motion compensated frame interpolation in high-definition video processing," in *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pp. 1633–1636, May 2009.

[67] Li, R., Zeng, B., and Liou, M. L., "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 438–442, Aug. 1994.

[68] Lim, K. P., Das, A., and Chong, M. N., "Estimation of occlusion and dense motion fields in a bidirectional bayesian framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 712–718, May 2002.

[69] Ling, Y., Wang, J., Liu, Y., and Zhang, W., "A novel spatial and temporal correlation integrated based motion-compensated interpolation for frame rate up-conversion," *IEEE Trans. Consum. Electron.*, vol. 54, pp. 863–869, May 2008.

[70] Lu, Z., Lin, W., Yang, X., Ong, E., and Yao, S., "Modeling visual attention's modulatory aftereffects on visual sensitivity and quality evaluation," *IEEE Trans. Image Process.*, vol. 14, pp. 1928 –1942, Nov. 2005.

[71] Luessi, M. and Katsaggelos, A., "Efficient motion compensated frame rate upconversion using multiple interpolations and median filtering," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pp. 373–376, Nov. 2009.

[72] Mertens, M. J. W. and de Haan, G., "Motion vector field improvement for picture rate conversion with reduced halo," in *VCIP*, pp. 352–362, 2001.

[73] Munoz-Jimenez, V., Mokraoui-Zergainoh, A., and Astruc, J.-P., "Bidirectional motion estimation approach using warping mesh combined to frame interpolation," in *Signal Processing and Information Technology, 2008. ISSPIT 2008. IEEE International Symposium on*, pp. 249–253, Dec. 2008.

[74] Nie, Y., Kong, H.-S., Vetro, A., Sun, H., and Barner, K., "Fast adaptive fuzzy post-filtering for coding artifacts removal in interlaced video," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, vol. 2, pp. 993–996, 2005.

[75] Nieweglowski, J., Moisala, T., and Haavisto, P., "Motion compensated video sequence interpolation using digital image warping," in *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, vol. V, pp. 205–208, Apr. 1994.

[76] Ojo, O. and de Haan, G., "Robust motion-compensated video upconversion," *Consumer Electronics, IEEE Transactions on*, vol. 43, pp. 1045–1056, Nov. 1997.

[77] Ong, E., Wang, H., and Xue, P., "Video coding based on true motion estimation," vol. 3, pp. 409–412, Apr. 2003.

[78] Palmer, S. E., *Vision Science: Photons to Phenomenology*. Cambridge, MA: MIT Press, 1999.

[79] Pang, Z. and Tan, H., "LCD motion blur blind modeling and analysis," in *Multimedia Technology (ICMT), 2010 International Conference on*, pp. 1–4, Oct. 2010.

[80] Po, L.-M. and Ma, W.-C., "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 313–317, June 1996.

[81] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 3 ed., August 2007.

[82] Richter, M. M., Dolar, C., and Schroder, H., "Coding artifact reduction by temporal filtering," in *Consumer Electronics, 2009. ISCE '09. IEEE 13th International Symposium on*, (Kyoto), pp. 6–10, May 2009.

[83] Sohn, Y. W. and Kang, M. G., "Block-based recursive motion filtering for preserving true motion vectors in time-varying texture objects," *International Journal of Imaging Systems and Technology*, vol. 18, pp. 265–275, Oct. 2008.

[84] Song, H., Men, A., and Shi, G., "A method for halo artifact reduction in MEMC," in *Consumer Electronics, 2009. ICCE '09. Digest of Technical Papers International Conference on*, pp. 1–2, Jan. 2009.

[85] Souk, J. and Lee, J., "Recent picture quality enhancement technology based on human visual perception in LCD TVs," *IEEE/OSA J. Display Technol.*, vol. 3, pp. 371–376, Dec. 2007.

[86] Stone, M. H., "The generalized Weierstrass approximation theorem," *Mathematics Magazine*, vol. 21, no. 4, pp. 167–184, 1948.

[87] Stone, M. H., "The generalized Weierstrass approximation theorem," *Mathematics Magazine*, vol. 21, no. 5, pp. 237–254, 1948.

[88] Sugiyama, K., Fujita, M., Yoshida, T., and Hangai, S., "Motion compensated frame rate up-conversion for low frame rate video," in *Visual Content Processing and Representation* (Atzori, L., Giusto, D., Leonardi, R., and Pereira, F., eds.), vol. 3893 of *Lecture Notes in Computer Science*, pp. 170–178, Springer Berlin / Heidelberg, 2006.

[89] Tai, S.-C., Chen, Y.-R., Huang, Z.-B., and Wang, C.-C., "A multi-pass true motion estimation scheme with motion vector propagation for frame rate up-conversion applications," *IEEE/OSA J. Display Technol.*, vol. 4, pp. 188–197, July 2008.

[90] Tekalp, A. M., *Digital video processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1995.

[91] Tham, J. Y., Ranganath, S., Ranganath, M., and Kassim, A., "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 369–377, Aug. 1998.

[92] Thoma, R. and Bierling, M., "Motion compensating interpolation considering covered and uncovered background," *Signal Processing: Image Communication*, vol. 1, pp. 191–212, Oct. 1989.

[93] Tu, S.-F., Au, O., Wu, Y., Luo, E., and Yeung, C.-H., "A novel framework for frame rate up conversion by predictive variable block-size motion estimated optical flow," in *Image and Signal Processing, 2009. CISP '09. 2nd International Congress on*, pp. 1–5, Oct. 2009.

[94] Tuan, J.-C., Chang, T.-S., and Jen, C.-W., "On the data reuse and memory bandwidth analysis for full-search block-matching vlsi architecture," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 61–72, Jan. 2002.

[95] Wahby, M. A., Mostafa, K., and Darwish, A. M., "DCT-based MPEG-2 programmable coprocessor," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* (S. Panchanathan, V. M. Bove, &

S. I. Sudharsanan, ed.), vol. 4674 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pp. 14–20, Dec. 2001.

[96] Wang, C., Zhang, L., He, Y., and Tan, Y.-P., "Frame rate up-conversion using trilateral filtering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, pp. 886–893, June 2010.

[97] Wang, D. and Lauzon, D., "Hybrid algorithm for estimating true motion fields," *Optical Engineering*, vol. 39, no. 11, pp. 2876–2881, 2000.

[98] Wang, J., Wang, D., and Zhang, W., "Temporal compensated motion estimation with simple block-based prediction," *IEEE Trans. Broadcast.*, vol. 49, pp. 241–248, Sept. 2003.

[99] Wang, Y., Ma, Z., and Ou, Y.-F., "Modeling rate and perceptual quality of scalable video as functions of quantization and frame rate and its application in scalable video adaptation," in *Packet Video Workshop, 2009. PV 2009. 17th International*, pp. 1–9, May 2009.

[100] Wang, Y., Zhang, Y.-q., and Ostermann, J., *Video Processing and Communications*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1st ed., 2001.

[101] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E., "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, pp. 600–612, Apr. 2004.

[102] Wedi, T., "Adaptive interpolation filters and high-resolution displacements for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, pp. 484–491, Apr. 2006.

[103] Wittebrood, R., de Haan, G., and Lodder, R., "Tackling occlusion in scan rate conversion systems," in *Consumer Electronics, 2003. ICCE. 2003 IEEE International Conference on*, pp. 344–345, June 2003.

[104] Yang, K.-C., Huang, A.-M., Nguyen, T. Q., Guest, C. C., and Das, P. K., "A new objective quality metric for frame interpolation used in video compression," *IEEE Trans. Broadcast.*, vol. 54, pp. 680–690, Sept. 2008.

[105] Zhao, L. and Zhou, Z., "A new algorithm for motion-compensated frame interpolation," in *Circuits and Systems, 1993., ISCAS '93, 1993 IEEE International Symposium on*, vol. 1, pp. 9–12, May 1993.

[106] Zhu, C., Lin, X., Chau, and Po, L.-M., "Enhanced hexagonal search for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, pp. 1210–1214, Oct. 2004.

[107] Zhu, C., Lin, X., and Chau, L.-P., "Hexagon-based search pattern for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 5, pp. 349–355, 2002.

[108] Zhu, S. and Ma, K.-K., "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. Image Process.*, vol. 9, pp. 287–290, Feb. 2000.

VITA

Salih Dikbaş received his B.S. degree in electrical and electronics engineering in 1996 from Middle East Technical University, Ankara, Turkey, and the M.S. degree in electrical and computer engineering in 1998 from Clemson University, Clemson, SC. He has completed his Ph.D. research under the supervision of Prof. Yücel Altunbaşak at the Georgia Institute of Technology, Atlanta, GA. His research interests include digital signal processing, digital image/video processing and compression. He is currently employed as a Systems Engineer in the Video Architecture & IP, DSPS department of Texas Instruments Inc., Dallas, TX.